Interplay of adversarial robustness and generalization in deep convolutional models

Ivan Grubišić
Mentor: Siniša Šegvić

Faculty of Electrical Engineering and Computing
Department of Electronics, Microelectronics, Computer and Intelligent Systems

Content

Outline

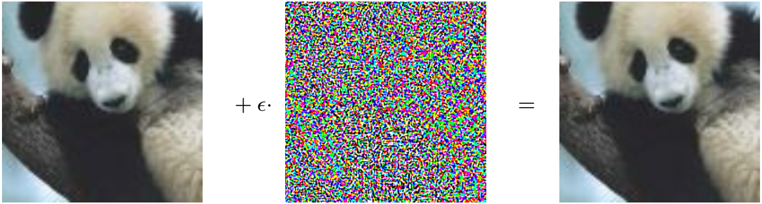
1. Nonrobustness of machine learning algorithms
2. Adversarial example definitions
3. Properties of adversarial examples
4. Attacks and defenses
5. Adversarial robustness and generalization
6. Conclusion

Nonrobustness of machine learning algorithms

- Current state-of-the-art machine learning algorithms do not work well with **domain-shifted, out-of-distribution and inputs crafted to fool them** and they often make **overconfident predictions** [Engstrom et al. (2017), Ganin and Lempitsky (2015), Hendrycks and Dietterich (2019), Hendrycks and Gimpel (2016), Nguyen et al. (2015), and Szegedy et al. (2013)].
- It is even possible to slightly, even impreceptibly, modify an input (e.g. image) to generate an adversarial example and cause a misprediction.
- This is not limited to complex deep models
- This indicates that current algorithms **perform well without actually understanding data** (in a way similar to humans).

Nonrobustness of machine learning algorithms

- A single small gradient descent step on an image increasing the loss is often enough to fool a classifier [Goodfellow et al. (2014)].
- Sometimes a single pixel can change the prediction [Su et al. (2017)].



$$\boldsymbol{x} + \epsilon \cdot \operatorname{sgn}(\nabla_{\boldsymbol{x}} L(y, h(\boldsymbol{x}))) = \tilde{\boldsymbol{x}}$$

panda (0.577) | *gibbon* (0.993)

Figure 1: Generation of an adversarial example with FGSM, a single step attack. Italic words and numbers represent classes and confidences. The images are from Goodfellow et al. (2014).

Adversarial robustness and generalization

- Evidence suggests that there is a **trade-off between robustness and generalization** with current algorithms [Madry et al. (2017), Su et al. (2018), and Tsipras et al. (2018)].
- The trade-off is **counter-intuitive** because **a hypothesis which optimally generalizes would have no adversarial examples** (and an optimal hypothesis exists given a data distribution).
- The question remains whether it is achievable regarding computational and data limitations to implement such algorithms.

Outline

Adversarial example definitions

Definition (practical adversarial example)

A practical adversarial example is an input for which the following holds:

1. It is **close** to an input x with a correct prediction.
2. The **hypothesis** produces a **different prediction** than for x.

Definition (adversarial example)

An adversarial example is an input for which the following holds:

1. It is **close** to an input with a correct prediction.
2. The **hypothesis** produces a **misprediction**.

Adversarial example definitions

- The set of adversarial examples is a function of the **hypothesis**, a **neighbourhood function**, the **input data distribution**, and either
  - a reference **input** (first definition) or
  - the **true hypothesis** (second definition).
- The practical definition is
  - **inconsistent** – examples close to class boundaries can be both adversarial and correctly classified depending on the reference, and
  - **practical for generating adversarial examples and robustnes evaluation**.
- The second definition is
  - **impractical** – it requires knowing the true hypothesis,
  - **consistent** – the true hypothesis has no adversarial examples, and
  - **helpful for achieving the goal of both robustness and generalization**.

Outline

Properties of adversarial examples

- Adversarial examples are **close to clean inputs and rare**, i.e. hard to find by randomly sampling the L^p neighbourhood [Szegedy et al. (2013)].
- The neighbourhood of an input contains adversarial examples classified in different classes, i.e. an input is **close to many class boundaries** of the learned hypothesis.
- Knowing the locally **linear** behaviour of the hypothesis is often enough to generate an adversarial example [Goodfellow et al. (2014)].
- Adversarial examples **generalize across algorithms and datasets**, i.e. an adversarial example of one model is often also an adversarial example of some other trained model [Liu et al. (2017), Papernot et al. (2016), Szegedy et al. (2013), and Tramèr et al. (2017)].

Properties of adversarial examples

- Tanay and Griffin (2016) hypothesize that adversarial examples might be occurring along **low-variance directions of the data** and that robustness could be improved with regularization.
- Gilmer et al. (2018) hypothesize that the existence of adversarial examples could be a naturally occurring result of the **geometry of high-dimensional data manifolds**.
- Ilyas et al. (2019) and Tsipras et al. (2018) hypothesize that adversarial examples exist because classifiers rely on **highly predictive but brittle (nonrobust) features**. Ilyas et al. (2019) provide good evidence.

Properties of adversarial examples

- (Some) generative models are also vulnerable to adversarial attacks as well [Goodfellow et al. (2014) and Kos et al. (2018)]. Figure 2 shows adversarial examples on a VAE-GAN.

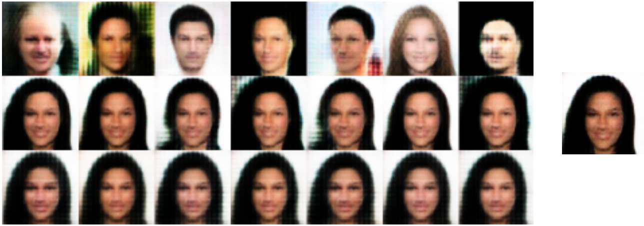


Figure 2: Reconstruction outputs for targeted attacks on a VAE-GAN from Kos et al. (2018). Rows represent reconstructions of original images (top), adversarial examples generated using an attack in latent space (middle) and a VAE-loss attack (bottom). The target reconstruction is on the right.

Properties of adversarial examples

- Adversarial examples of **robust classifiers** are truly **ambigous** to humans [Li (2018) and Tsipras et al. (2018)], which suggests that they **understand data** much better. The semantic meaningfulness of adversarial examples of robust hypotheses is illustrated in figures 3, 4, and 5.

Properties of adversarial examples

Figure 3: Cherry-picked original images and adversarial examples generated with a large perturbation using an iterative non-targeted attack on an adversarially trained Restricted ImageNet classifier from Tsipras et al. (2018).

Properties of adversarial examples

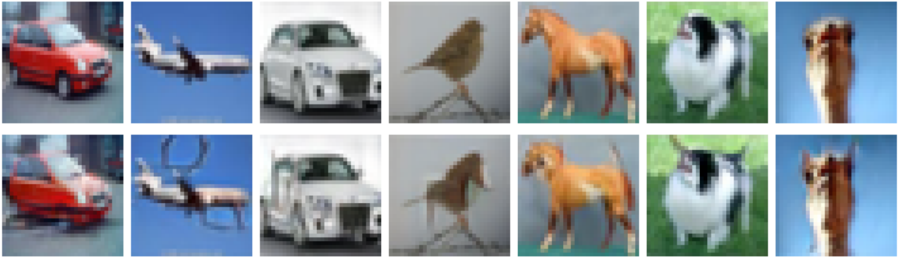


Figure 4: Cherry-picked clean images (top) and adversarial examples (bottom) generated using an iterative L^2-bounded attack on a CIFAR-10 classifier. The predicted classes for the bottom row are *ship*,*deer*,*truck*,*horse*,*dog*,*cat*,*cat*. Adapted from [Rony et al. (2018)].

Properties of adversarial examples

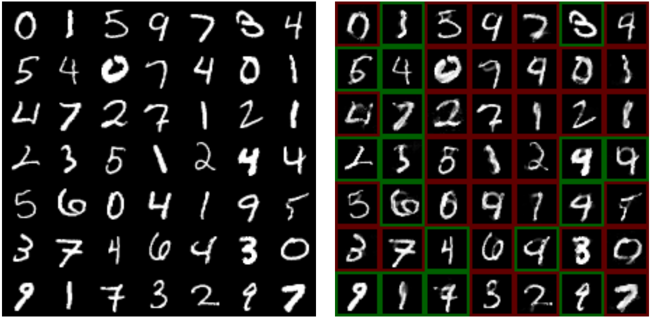


Figure 5: Clean images (left) and adversarial examples generated using an iterative non-targeted attack on a generative MNIST classifier with the factorization $\mathrm{p}(\boldsymbol{z})\,\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{z})\,\mathrm{p}(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{y})$ (right) from Li (2018). The adversarial examples marked in green are successful.

Outline

Finding adversarial examples

- Let $\mathbb{X}$ be the input space, and $d \in (\mathbb{X} \times \mathbb{X} \to \mathbb{R}^+)$ a **distance function**. The **neighbourhood** of an example $\boldsymbol{x}$ can be $B_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}' : d(\boldsymbol{x}', \boldsymbol{x}) \leq \epsilon\}$, where ϵ is the maximum distance.
- Ideally, the neighbourhood of an example $\boldsymbol{x}$ should be the set of **perceptually similar** examples that all belong to the same class as $\boldsymbol{x}$, but it requires knowing the true hypothesis.
- A common choice for distance d is L^p distance with $p \in \{1, 2, \infty\}$.
- Finding an adversarial example can be defined as a constrained optimization problem of **maximizing some loss with respect to the input** with a constraint of a neighbourhood $B_\epsilon(\boldsymbol{x})$:

$$\tilde{\boldsymbol{x}} = \underset{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})}{\arg\max}\, L(y, h(\boldsymbol{x}')), \tag{1}$$

where y is the true label, h the hypothesis, and L the loss function.

Finding adversarial examples

- An objective can also be to find **the closest adversarial example** [Moosavi-Dezfooli et al. (2016)]:

$$\tilde{\boldsymbol{x}} = \underset{\boldsymbol{x}' :\, \boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}) \wedge \hat{h}(\boldsymbol{x}') \neq y}{\arg\min} d(\boldsymbol{x}', \boldsymbol{x}), \tag{2}$$

where $\hat{h}(\boldsymbol{x}) \coloneqq \arg\max_y h(\boldsymbol{x})_{[y]}$.

- There are also **targeted attacks**, where the objective is to get an adversarial example that is classified to some target class. Targeted attack objectives corresponding to equations (1) and (2) are:

$$\tilde{\boldsymbol{x}} = \underset{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})}{\arg\min} L(y_{\mathsf{a}}, h(\boldsymbol{x}')), \tag{3}$$

$$\tilde{\boldsymbol{x}} = \underset{\boldsymbol{x}' :\, \boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}) \wedge \hat{h}(\boldsymbol{x}') = y_{\mathsf{a}}}{\arg\min} d(\boldsymbol{x}', \boldsymbol{x}), \tag{4}$$

where y_{a} denotes the adversarial target label.

Finding adversarial examples

- Non-targeted adversarial examples can also be generated by using the prediction instead of the true label in the loss, resulting in **virtual adversarial examples**.
- Kurakin et al. (2016) and Miyato et al. (2017) propose the following attack objective for use in semi-supervised learning:

$$\tilde{\boldsymbol{x}} = \underset{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})}{\arg\min}\, D((\underline{y} \mid \boldsymbol{x}, \boldsymbol{\theta}), (\underline{y} \mid \underline{\boldsymbol{x}} = \boldsymbol{x}', \boldsymbol{\theta})), \tag{5}$$

where D is some distribution distance function.

Common attacks

- General constrained optimization algorithms as well as more specific ones can be used to generate adversarial examples.
- Some common atacks are:
  - Box-constrained L-BFGS – Szegedy et al. (2013) propose to minimize $c\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2^2 + L(y, h(\tilde{\boldsymbol{x}}))$ with the constraint $\tilde{\boldsymbol{x}} \in [0, 1]$ with L-BFGS, a quasi-Newton optimization method.
  - Fast gradient sign method (FGSM) – an attack proposed by Goodfellow et al. (2014) that requires a single gradient computation:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \epsilon \nabla_{\boldsymbol{x}} L(y, h(\boldsymbol{x})). \tag{6}$$

Common attacks

- Projected gradient descent (PGD) [Madry et al. (2017)][1] – an iterative gradient-based algorithm with random initialization [Madry et al. (2017)] of the perturbation from within $B_\epsilon(\boldsymbol{x})$ at the start and steps in the direction of the gradient sign:

$$\tilde{\boldsymbol{x}}_i = \Pi_{B_\epsilon(\boldsymbol{x})}\big(\tilde{\boldsymbol{x}}_{i-1} + \alpha\,\mathrm{sgn}\big(\nabla_{\tilde{\boldsymbol{x}}_{i-1}} L(y, h(\tilde{\boldsymbol{x}}_{i-1}))\big)\big). \tag{7}$$

  α is the step size, $\Pi_{B_\epsilon(\boldsymbol{x})}$ is the projection into the L^p ϵ-ball around $\boldsymbol{x}$.
- Carlini-Wagner (CW) attacks – Carlini and Wagner (2017b) propose attacks with similar minimal perturbation objectives as Szegedy et al. (2013) . They modify the loss and they introduce change of variables $\boldsymbol{\delta} = \frac{1}{2}(\tanh(\boldsymbol{w}) + \mathbf{1}) - \boldsymbol{x}$ to limit the perturbation $\boldsymbol{\delta}$ to $[0, 1]$.

- The CW and PGD attacks are currently some of the strongest attacks, suitable for robustness evaluation.

[1] Equialent to BIM [Kurakin et al. (2016)] up to random initialization.

Improving adversarial robustness

- There are different defenses, most of which have been shown to be nonrobust, but had appeared robust because of deficiencies in robustness evaluation [Athalye et al. (2018), Carlini and Wagner (2017a), Carlini and Wagner (2017b), and Uesato et al. (2018)].
- Some approaches use **generative models** to approximately **project inputs to a learned data manifold** (e.g. Samangouei et al. (2018))
- Some are based on **limiting the Lipschitz constant** of the model to limit sensitivity to small input perturbations by regularization and model modification (e.g. Qian and Wegman (2018)), some research is looking into ways of **guaranteeing robustness** (e.g. Cohen et al. (2019)).
- The defense currently believed to be most effective according to Athalye et al. (2018) is **adversarial training** [Goodfellow et al. (2014)] with a strong iterative attack [Madry et al. (2017)].

Adversarial training and empirical adversarial risk

- Madry et al. (2017) define what can be called **empirical adversarial risk** by allowing the worst-case attack to modify each input:

$$R_{\mathsf{EA}}(h, \mathbb{D}) := \mathop{\mathbf{E}}_{(\boldsymbol{x},y)\sim p_{\mathbb{D}}} \left(\max_{\tilde{\boldsymbol{x}} \in B_\epsilon(\boldsymbol{x})} L(y, h(\tilde{\boldsymbol{x}})) \right). \tag{8}$$

- They propose PGD for the attack during training and PGD with many iterations to get a better upper bound on robustness.
- Adversarially trained models are **not robust to stronger attacks** than those used for training [Schott et al. (2018)]. Because of generating adversarial examples using non-ideal L^p distance, **performance is affected** [Madry et al. (2017) and Tsipras et al. (2018)] and there can exist **invariance-based adversarial examples** [Jacobsen et al. (2019)].

Outline

A trade-off between robustness and generalization

- Madry et al. (2017), Su et al. (2017), and Tsipras et al. (2018) and others have empirically observed that adversarial robustness with current algorithms requires **more capacity** and **negatively affects generalization**.
- Su et al. (2017) observe that older convolutional architectures with no shortcut connections seem to be inherently more robust than better performing architectures with standard training.
- Tsipras et al. (2018), based on the practical definition of an adversarial example, theoretically demonstrate an aspect of the trade-off.
- Another cause of performance drop suggested by them is that salient features might be **harder to learn** and that algorithms rely on **highly predictive but nonrobust** features.

Nonrobust features

- Ilyas et al. (2019) give good experimental support for the hypothesis on reliance on highly-predictive nonrobust features.
- In one experiment, they
    1. construct a training set $\mathbb{D}_{\mathsf{NR}}$ where the only **useful features are nonrobust features** by generating **adversarial examples** for a standardly trained classifier and **relabeling** them with target labels,
    2. **train a new classifier on the nonrobust dataset** $\mathbb{D}_{\mathsf{NR}}$,
    3. test the new classifier on the original test set, where it achieves **performance close to the original classifier** and **lower robustness**.
- In another experiment, they
    1. construct a training set $\mathbb{D}_{\mathsf{R}}$ with removed **nonrobust features** using a robust classifier,
    2. **train a new classifier on the robust dataset** $\mathbb{D}_{\mathsf{R}}$,
    3. test the new classifier on the original test set, where it achieves a bit **lower performance** and **significantly higher robustness**.

Nonrobust features

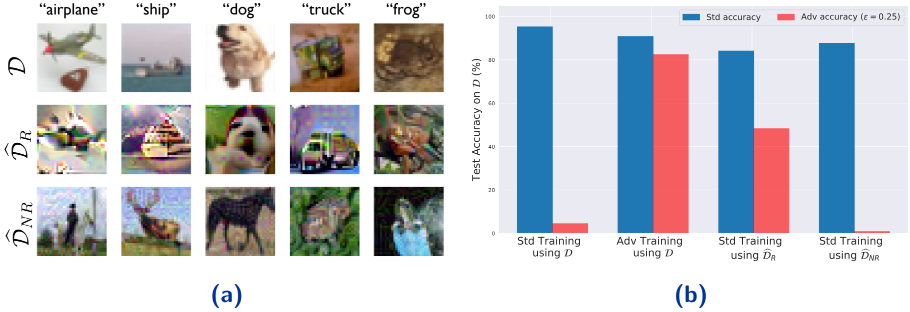


Figure 6: (a) Random samples from variants of the CIFAR-10 training set: the **original** training set; the **robust training set** $\mathbb{D}_\mathrm{R}$, with features used by a robust model; and the **nonrobust training set** $\mathbb{D}_\mathrm{NR}$, with features relevant to a standard model. (b) Standard and robust accuracy on the CIFAR-10 test set ($\mathbb{D}$) for models trained with standard training, adversarial training, and standard training on datasets $\mathbb{D}_\mathrm{R}$ (robust) and $\mathbb{D}_\mathrm{NR}$ (nonrobust). Adapted from Ilyas et al. (2019).

Training with on-manifold adversarial examples

- Stutz et al. (2018) challenge the hypothesis that there is a fundamental trade-off between robustness and generalization.
- They hypothesize that most adversarial examples come from directions orthogonal to the learned class manifolds and that training with adversarial examples (as per a definition similar to definition 2) limited to the known or learned class manifolds (on-manifold adversarial examples) can improve generalization.
- Experiments with a synthetic dataset with known class-invariant transformations and datasets with small images support the hypothesis that generalization can be improved with adversarial training with on-manifold adversarial examples.

Training with on-manifold adversarial examples

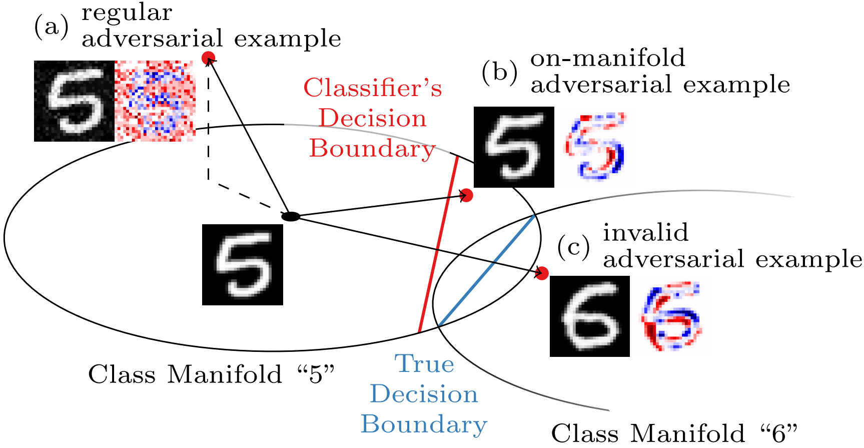


Figure 7: An illustration by Stutz et al. (2018) of class manifolds (classes "5" and "6") with a regular (off-manifold) adversarial example and an on-manifold adversarial example.

Training with on-manifold adversarial examples

- In one of the experiments Stutz et al. (2018) construct a synthetic dataset with a known manifold (geometric transformations of letters) in order to be able to generate exactly on-manifold adversarial examples by modifying parameters of the geometric transformations. With this dataset, they succeed in improving generalization and on-manifold robustness[2] with adversarial training.
- In other experiments they use EMNIST [Cohen et al. (2017)], FashionMNIST [Xiao et al. (2017)] and CelebA [Liu et al. (2015)]. In order to better approximate class manifolds and disable leaving the manifold of a class when an adversarial example is generated for adversarial training, they first train one variational autoencoder (VAE-GAN) per class. They perform training and evaluation analogously to the experiment with synthetic data by allowing the

Training with on-manifold adversarial examples

attack to perturb the latent representation of the autoencoder corresponding to the correct class. They measure positive correlation between robustness to on-manifold adversarial examples and generalization. They observe worse quality of on-manifold adversarial examples for the more complex dataset CelebA due to worse approximation quality of their VAE-GAN-s.

[2] By the authors' definition of an adversarial example, which is similar to the consistent definition (definition 2), except for that there is no closeness constraint, making it equivalent to the definition of a misclassified example, on-manifold robustness essentially boils down to generalization.

Outline

Conclusion

- Some recent results [Stutz et al. (2018)] suggest that finding ways of improving both robustness generalization might be an interesting research direction to explore.

References

Athalye, Anish, Nicholas Carlini, and David Wagner (2018). "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmssan, Stockholm Sweden: PMLR, pp. 274–283. URL: http://proceedings.mlr.press/v80/athalye18a.html.

Brown, Tom B. et al. (2018). "Unrestricted Adversarial Examples". In: *CoRR* abs/1809.08352. arXiv: 1809.08352. URL: http://arxiv.org/abs/1809.08352.

Carlini, Nicholas and David Wagner (2017a). "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. AISec '17. Dallas, Texas, USA: ACM, pp. 3–14. ISBN: 978-1-4503-5202-4. DOI: 10.1145/3128572.3140444. URL: http://doi.acm.org/10.1145/3128572.3140444.

Carlini, Nicholas and David A. Wagner (2017b). "Towards Evaluating the Robustness of Neural Networks". In: *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 39–57. DOI: 10.1109/SP.2017.49. URL: https://doi.org/10.1109/SP.2017.49.

Cohen, Gregory et al. (2017). "EMNIST: an extension of MNIST to handwritten letters". In: *arXiv preprint arXiv:1702.05373*.

References

Cohen, Jeremy, Elan Rosenfeld, and Zico Kolter (2019). "Certified Adversarial Robustness via Randomized Smoothing". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, pp. 1310–1320. URL: http://proceedings.mlr.press/v97/cohen19c.html.

Engstrom, Logan et al. (2017). "A Rotation and a Translation Suffice: Fooling CNNs with Simple Transformations". In: *CoRR* abs/1712.02779. arXiv: 1712.02779. URL: http://arxiv.org/abs/1712.02779.

Gal, Y. and L. Smith (June 2018). "Sufficient Conditions for Idealised Models to Have No Adversarial Examples: a Theoretical and Empirical Study with Bayesian Neural Networks". In: *ArXiv e-prints*. arXiv: 1806.00667 [stat.ML].

Ganin, Yaroslav and Victor Lempitsky (2015). "Unsupervised Domain Adaptation by Backpropagation". In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, pp. 1180–1189. URL: http://dl.acm.org/citation.cfm?id=3045118.3045244.

Gilmer, Justin et al. (2018). "Adversarial Spheres". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. URL: https://openreview.net/forum?id=SkthlLkPf.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy (2014). "Explaining and Harnessing Adversarial Examples". In: URL: http://arxiv.org/abs/1412.6572.

References

Hendrycks, Dan and Thomas G. Dietterich (2019). "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *CoRR* abs/1903.12261. arXiv: 1903.12261. URL: http://arxiv.org/abs/1903.12261.

Hendrycks, Dan and Kevin Gimpel (2016). "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *CoRR* abs/1610.02136. arXiv: 1610.02136. URL: http://arxiv.org/abs/1610.02136.

Ilyas, Andrew et al. (2019). *Adversarial Examples Are Not Bugs, They Are Features*. cite arxiv:1905.02175. URL: http://arxiv.org/abs/1905.02175.

Jacobsen, Jörn-Henrik et al. (2019). "Exploiting Excessive Invariance caused by Norm-Bounded Adversarial Robustness". In: *CoRR* abs/1903.10484. arXiv: 1903.10484. URL: http://arxiv.org/abs/1903.10484.

Kos, J., I. Fischer, and D. Song (2018). "Adversarial Examples for Generative Models". In: *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 36–42. DOI: 10.1109/SPW.2018.00014.

Krizhevsky, Alex (2009). *Learning multiple layers of features from tiny images*. Tech. rep.

Kurakin, Alexey, Ian J. Goodfellow, and Samy Bengio (2016). "Adversarial Machine Learning at Scale". In: *CoRR* abs/1611.01236. arXiv: 1611.01236. URL: http://arxiv.org/abs/1611.01236.

Li, Yingzhen (2018). "Are Generative Classifiers More Robust to Adversarial Attacks?" In: *CoRR* abs/1802.06552. arXiv: 1802.06552. URL: http://arxiv.org/abs/1802.06552.

References

Liu, Yanpei et al. (2017). "Delving into Transferable Adversarial Examples and Black-box Attacks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. URL: https://openreview.net/forum?id=Sys6GJqxl.

Liu, Ziwei et al. (Dec. 2015). "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*.

Madry, Aleksander et al. (2017). "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *CoRR* abs/1706.06083. arXiv: 1706.06083. URL: http://arxiv.org/abs/1706.06083.

Miyato, Takeru et al. (2017). "Virtual Adversarial Training: a Regularization Method for Supervised and Semi-supervised Learning". In: *CoRR* abs/1704.03976. arXiv: 1704.03976. URL: http://arxiv.org/abs/1704.03976.

Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard (2016). "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks". In: *CVPR*. IEEE Computer Society, pp. 2574–2582.

Nguyen, Anh Mai, Jason Yosinski, and Jeff Clune (2015). "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.". In: *CVPR*. IEEE Computer Society, pp. 427–436. ISBN: 978-1-4673-6964-0. URL: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2015.html#NguyenYC15.

References

Papernot, Nicolas, Patrick D. McDaniel, and Ian J. Goodfellow (2016). "Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples.". In: *CoRR* abs/1605.07277. URL: http://dblp.uni-trier.de/db/journals/corr/corr1605.html#PapernotMG16.

Qian, Haifeng and Mark N. Wegman (2018). "L2-Nonexpansive Neural Networks". In: *CoRR* abs/1802.07896. arXiv: 1802.07896. URL: http://arxiv.org/abs/1802.07896.

Rony, Jérôme et al. (2018). "Decoupling Direction and Norm for Efficient Gradient-Based L2 Adversarial Attacks and Defenses". In: *CoRR* abs/1811.09600. arXiv: 1811.09600. URL: http://arxiv.org/abs/1811.09600.

Samangouei, Pouya, Maya Kabkab, and Rama Chellappa (2018). "Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models". In: *CoRR* abs/1805.06605. arXiv: 1805.06605. URL: http://arxiv.org/abs/1805.06605.

Schott, Lukas et al. (2018). "Towards the first adversarially robust neural network model on MNIST". In: *CoRR* abs/1805.09190. arXiv: 1805.09190. URL: http://arxiv.org/abs/1805.09190.

Stutz, David, Matthias Hein, and Bernt Schiele (2018). "Disentangling Adversarial Robustness and Generalization". In: *CoRR* abs/1812.00740.

References

Su, Dong et al. (2018). "Is Robustness the Cost of Accuracy? - A Comprehensive Study on the Robustness of 18 Deep Image Classification Models". In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, pp. 644–661. DOI: 10.1007/978-3-030-01258-8_39. URL: https://doi.org/10.1007/978-3-030-01258-8_39.

Su, Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai (2017). "One pixel attack for fooling deep neural networks". In: *CoRR* abs/1710.08864. arXiv: 1710.08864. URL: http://arxiv.org/abs/1710.08864.

Szegedy, Christian et al. (2013). "Intriguing properties of neural networks". In: *CoRR* abs/1312.6199. arXiv: 1312.6199. URL: http://arxiv.org/abs/1312.6199.

Tanay, Thomas and Lewis D. Griffin (2016). "A Boundary Tilting Persepective on the Phenomenon of Adversarial Examples". In: *CoRR* abs/1608.07690. arXiv: 1608.07690. URL: http://arxiv.org/abs/1608.07690.

Tramèr, Florian et al. (2017). "The Space of Transferable Adversarial Examples". In: *CoRR* abs/1704.03453. arXiv: 1704.03453. URL: http://arxiv.org/abs/1704.03453.

Tsipras, Dimitris et al. (2018). *Robustness May Be at Odds with Accuracy*. cite arxiv:1805.12152. URL: http://arxiv.org/abs/1805.12152.

References

Uesato, Jonathan et al. (2018). “Adversarial Risk and the Dangers of Evaluating Against Weak Attacks”. In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 5032–5041. URL: http://proceedings.mlr.press/v80/uesato18a.html.

Xiao, Han, Kashif Rasul, and Roland Vollgraf (Aug. 28, 2017). “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms”. In: arXiv: cs.LG/1708.07747 [cs.LG].

Adversarial example definitions

- A common but imprecise definition of an adversarial example is *an input designed to fool a hypothesis into producing a misprediction*.
- Some broader definitions also consider **out-of-distribution** examples [Gal and Smith (2018)] or **any** inputs that fools the hypothesis [Brown et al. (2018)], but those will be not considered.

Robustness evaluation

- For adversarial training with weaker attacks, non-targeted attacks should be preferred due to **label leaking** [Kurakin et al. (2016)] where the learned classifier can overfit to adversarial examples and perform better on them than on natural examples, especially with attacks with a small number of iterations.
- For robustness evaluation with datasets that have many similar classes, non-targeted attacks can too easily fool the classifier and targeted attacks give more meaningful evaluation results [Athalye et al. (2018)].