Standard Adversarial Training

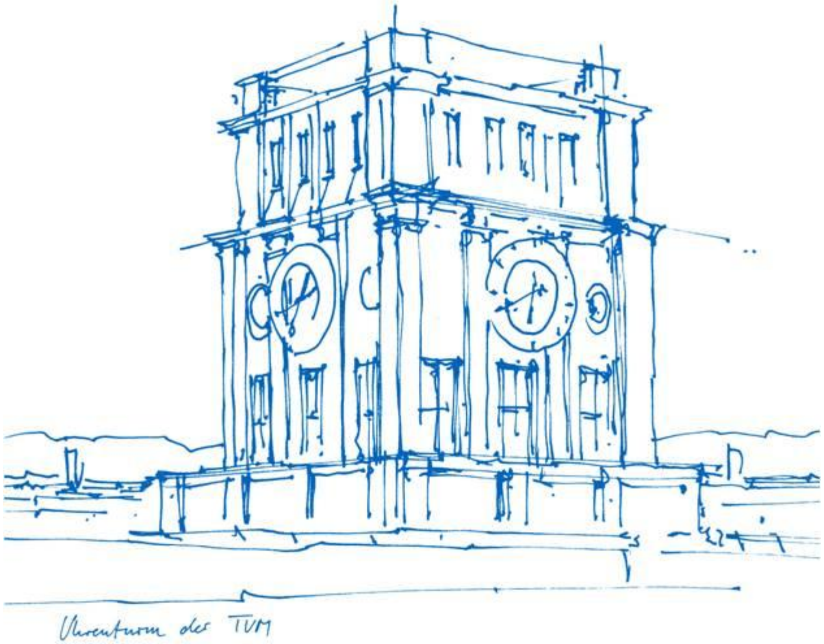
Theory and Review

Technical University Munich

Chair of Computer Science

Machine Learning Seminar, SS21

15.07.2021



Moritz Schüler

Adversarial Examples

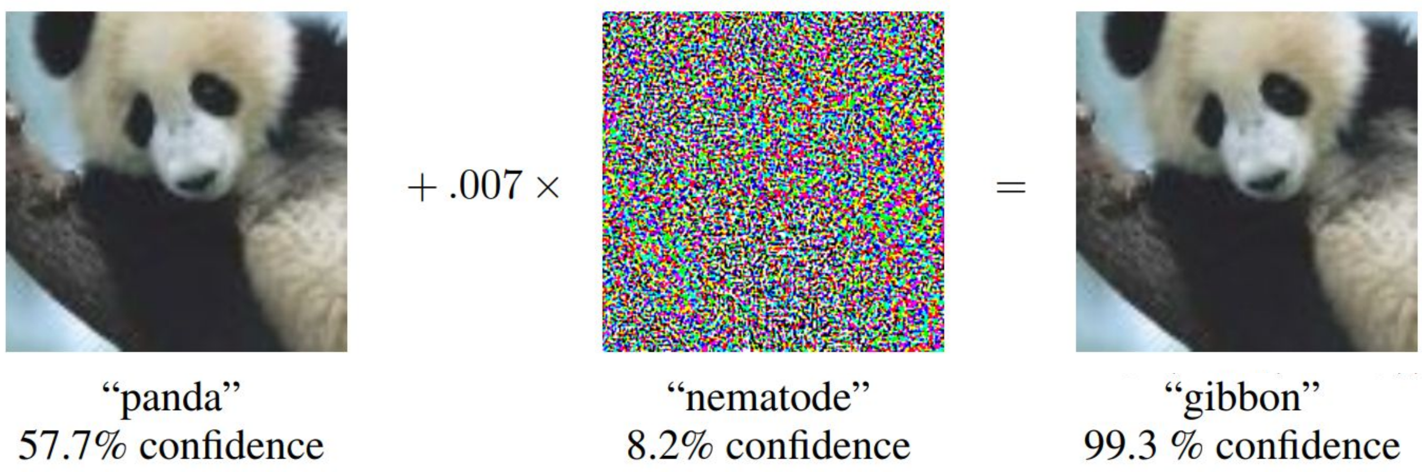


Fig 1: Adversarial example on GoogLeNet [1]

- Perturbing input s.t. it causes misclassification
- Here, perturbations constrained within Lp - ball

[1] Image taken from "Explaining and Harnessing Adversarial Examples" by Goodfellow et al.

Adversarial Examples

Fig 2: Adversarial example in real life applications, left: graffiti, right adversarial attack [2]

- Stop sign get classified as speed limit sign

[2] Image taken from “Robust physical-world attacks on deep learning visual classification” by Eykholt et al.

Adversarial Examples

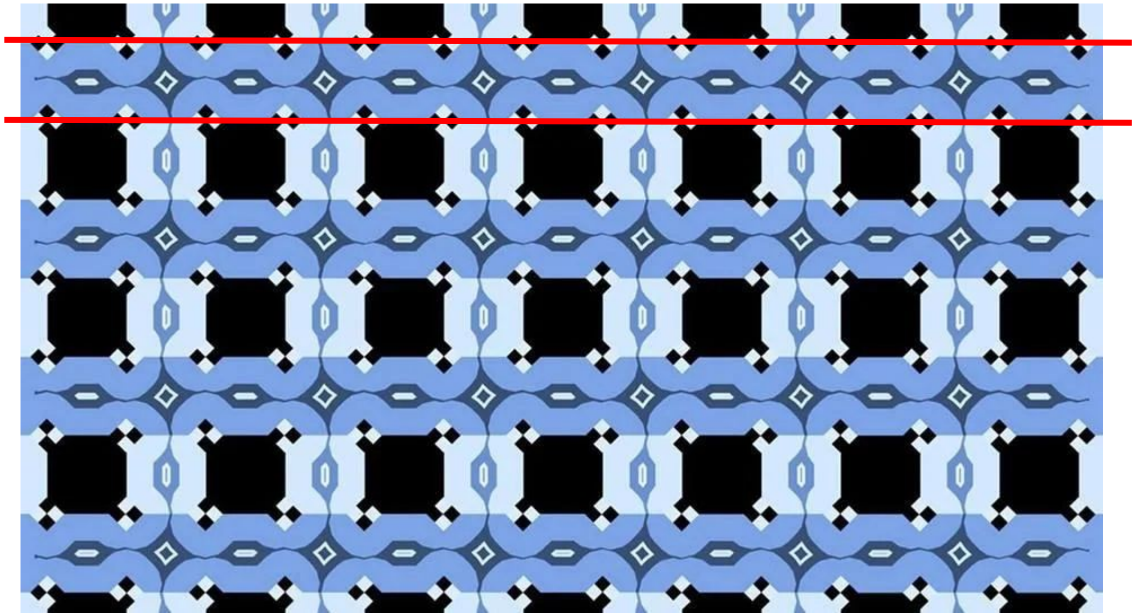


Fig 3: optical illusion for human brain [3]

- blue lines are straight and horizontal

[3] Image taken from Express

Why are neural networks prone to adversarial examples?

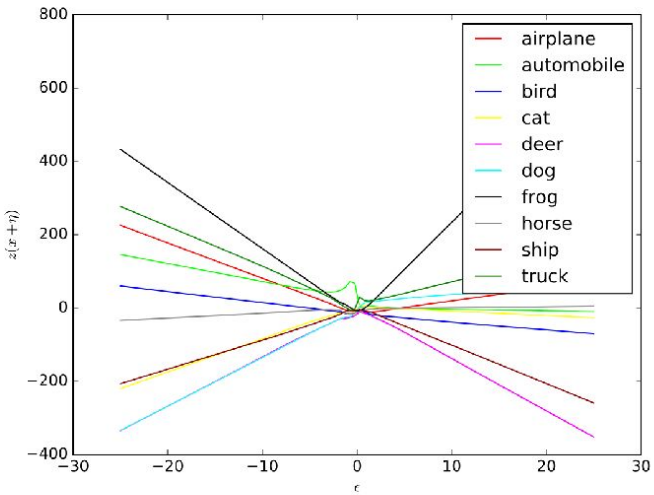


Fig 4: decision boundaries for a model trained on CIFAR10 [4]

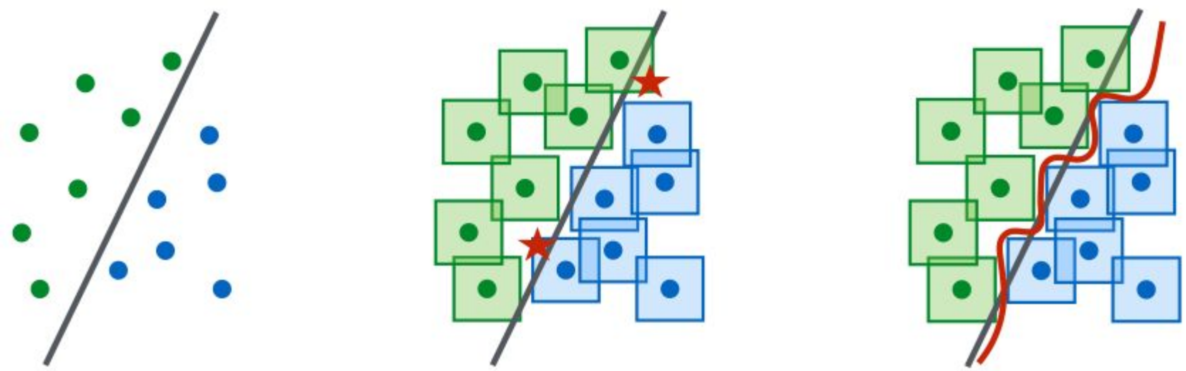


Fig 5: conceptual illustration of standard and adversarial decision boundaries [5]

- Excessive linearity of the decision boundaries

[4] Image taken from “Adversarial Examples and Adversarial Training” by Goodfellow et al.
[5] Image taken from “Towards Deep Learning Models Resistant to Adversarial Attacks” by Madry et al.

How to create adversarial examples?

- find perturbation δ that maximizes classification loss l

$$\max_{\delta \in \Delta} \quad l(f_\theta(x_i + \delta), y_i)$$

$$\Delta = \{\delta : ||\delta||_p \leq \epsilon\} \quad \text{with} \quad \epsilon > 0$$

- Δ being the threat model (bounded by an Lp - ball of size ε)

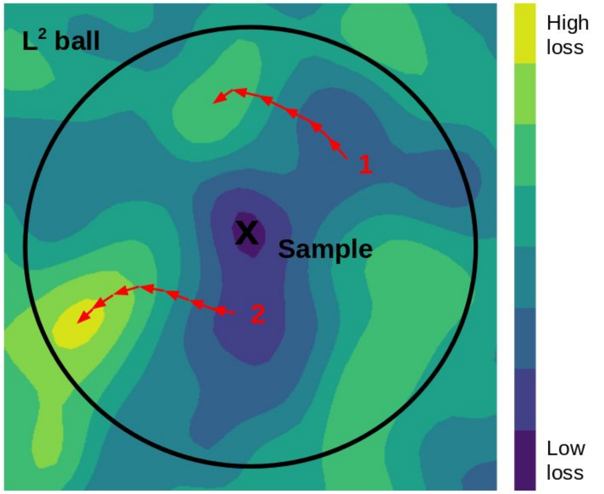
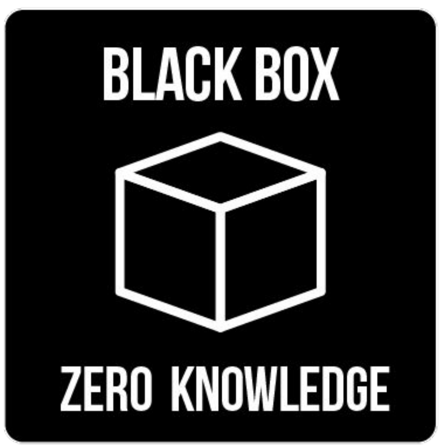


Figure 6: “The dynamics of a PGD attack in the loss landscape” [6]

- How? projected gradient ascent for x

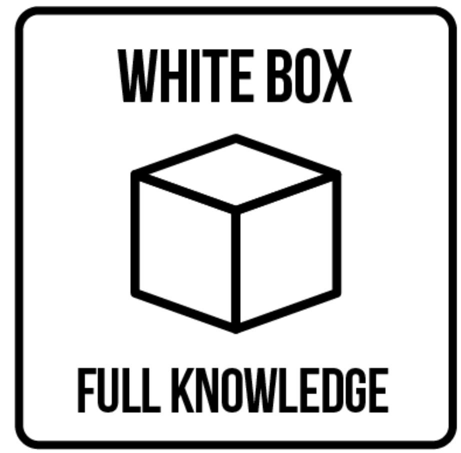
[6] Image taken from “Ignorance is Bliss: Adversarial Robustness by Design with LightOn OPUs” on Medium

Types of adversarial attacks



manual process, starting with random input

train substitute model, proceed like white box attack



gradient ascent to generate adversarial samples

Adversarial attacks are **model agnostic**!

Types of adversarial attacks

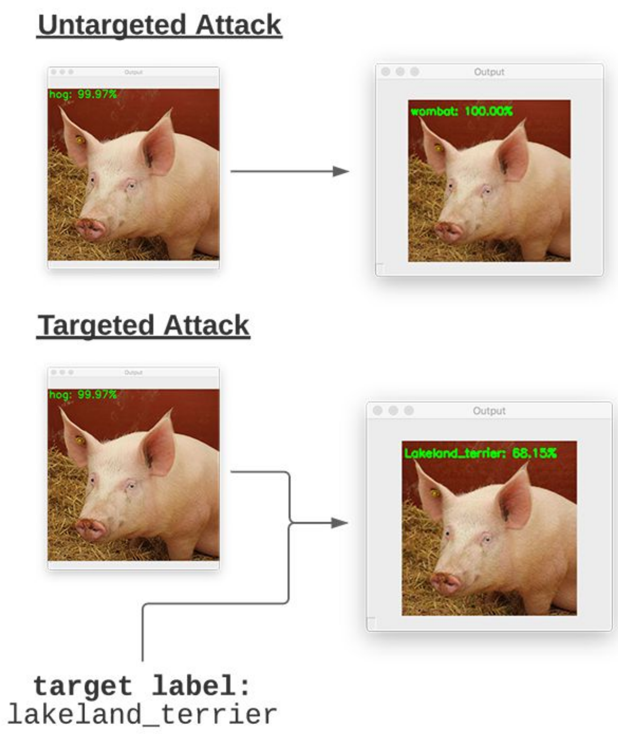


Figure 7: Examples of adversarial attacks[7]

- change label to some other class
- change label to given target class

[7] Image taken from “Targeted adversarial attacks with Keras and TensorFlow” on PyImageSearch

How to defend against adversarial attacks?

- basic idea: use adversarial examples for training

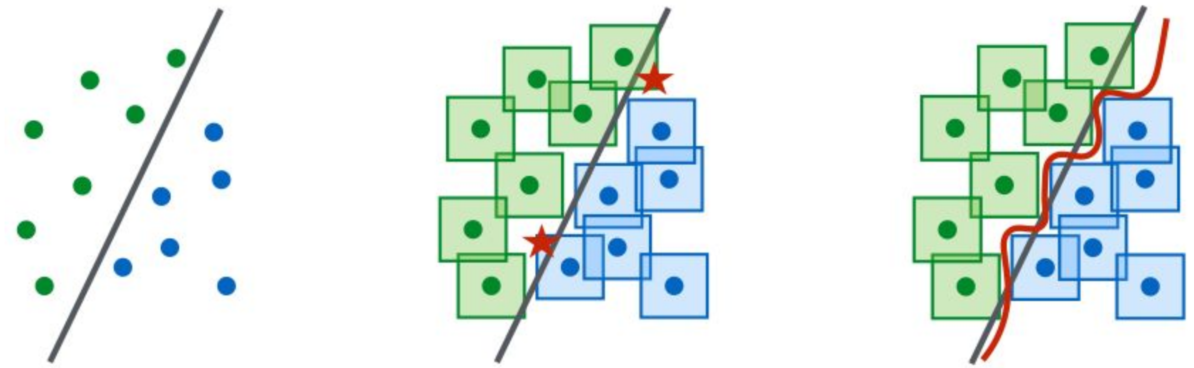


Fig 8: conceptual illustration of standard and adversarial decision boundaries [5]

[5] Image taken from “Towards Deep Learning Models Resistant to Adversarial Attacks” by Madry et al.

How to defend against adversarial attacks?

- basic idea: use adversarial examples for training

$$\min_\theta \sum_i \max_{\delta \in \Delta} \; l(f_\theta(x_i + \delta), y_i)$$
$$\Delta = \{\delta : ||\delta||_p \leq \epsilon\} \quad \text{with} \quad \epsilon > 0$$

- Challenge: how to calculate derivative?

Danskin's Theorem

The (sub)gradient of a function containing a max term can be found by taking the gradient at the point of the maximum δ^*.

$$\nabla_\theta \max_{||\delta|| \leq \epsilon} l(f_\theta(x_i + \delta), y_i) = \nabla_\theta l(f_\theta(x_i + \delta^*(x_i)), y_i)$$

- Requirements:
  - Convex loss function
  - only holds for exact maximum
- Limitations:
  - robustness depends on precision of maximum

Robust Optimization

- formulation as saddle point problem

$$\min_\theta \sum_i \max_{\delta \in \Delta} \; l(f_\theta(x_i + \delta), y_i)$$
$$\Delta = \{\delta : ||\delta||_p \leq \epsilon\} \quad \text{with} \quad \epsilon > 0$$

- robustness stems from strongness of attack model

Fast Gradient Sign Method (FGSM)

- take single step into gradient direction
- step size = ε to stay in Lp - ball

$$\tilde{x} = x + \epsilon \cdot sgn(\nabla_x l(\theta, x, y)))$$

- Fast, but not accurate

Multistep Projected Gradient Descent (K-PGD)

- take *k* smaller steps into gradient direction
- step size = α
- project back on Lp - ball if step outside

$$\tilde{x} = \Pi(x + \alpha \cdot sgn(\nabla_x l(\theta, x, y)))$$

setting $k = 1$ and $\alpha = \epsilon$ resembles FGSM

- more accurate, but slow

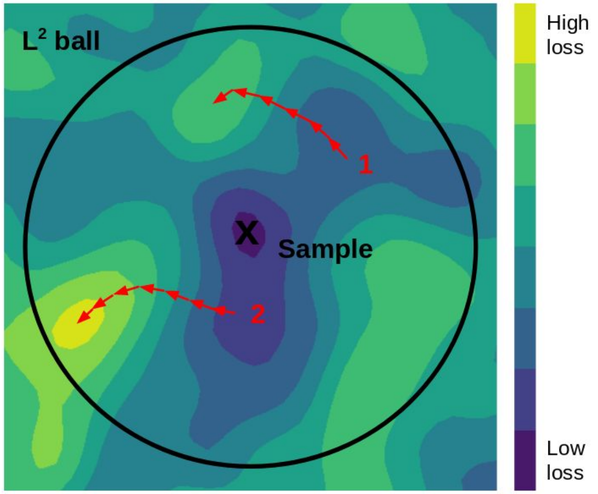


Figure 9: “The dynamics of a PGD attack in the loss landscape” [6]

[6] Image taken from “Ignorance is Bliss: Adversarial Robustness by Design with LightOn OPUs” on Medium

Multistep Projected Gradient Descent (K-PGD)

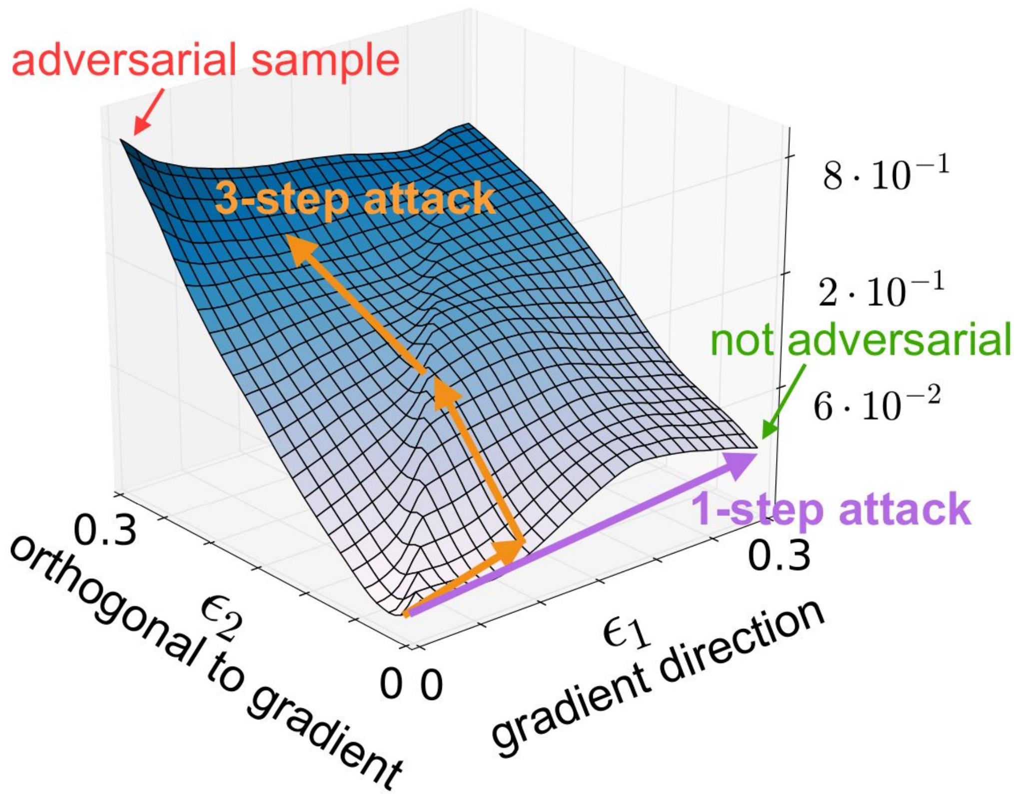


Figure 10: Comparison of FGSM and 3-PGD [8]

[8] Image taken from “Ensemble adversarial training: Attacks and defenses” by Tramer et al.

Advancements on FGSM

- Free Training:
  - re-use gradients from previous time step
  - mini batch replay
  - warm start with previous perturbation

- Fast Training:
  - re-use gradients from previous time step
  - random initialize perturbation

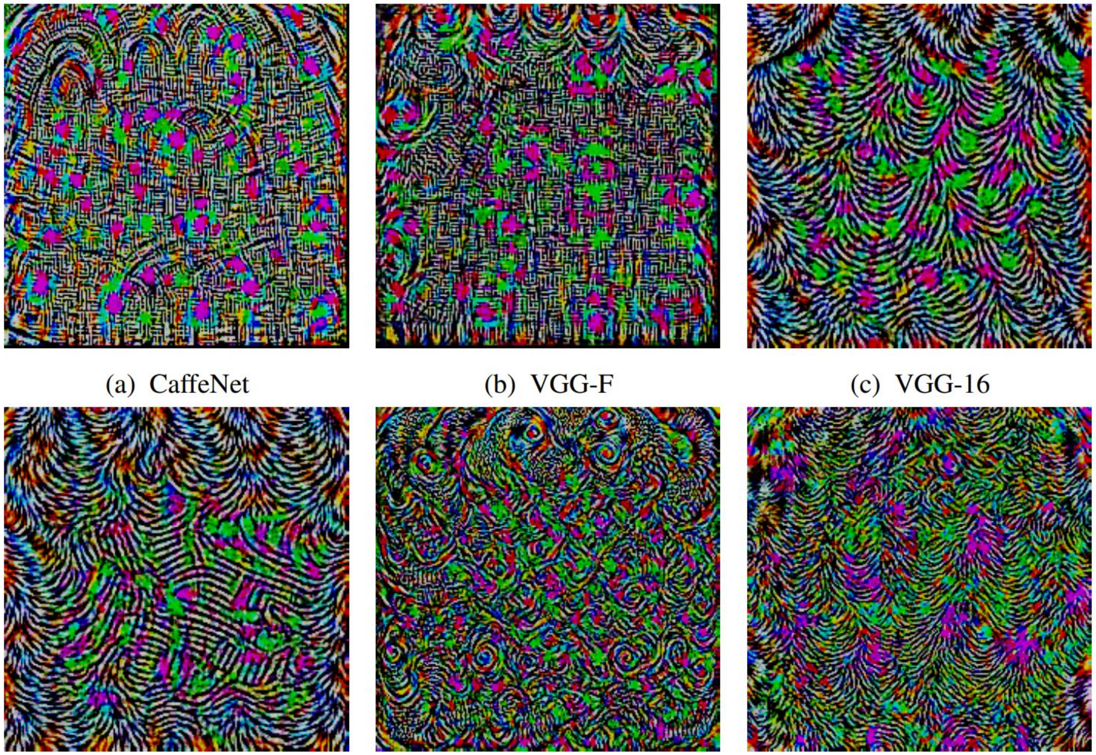
Universal Adversarial Training

- find a **single** perturbation that works on many inputs

$$\min_\theta \max_{\delta\in\Delta} \ \frac{1}{N}\sum_{i=1}^{N} \ \hat{l}\,(f_\theta(x_i+\delta), y_i)$$
$$\text{with} \quad \hat{l}\,(f_\theta(x_i+\delta), y_i) = \min\{l(f_\theta(x_i+\delta), y_i), \beta\}$$

- β bounds the loss from above to hinder a single sample to dominate the average loss
- advancement: relax formulation to allow perturbations **per class**

Universal Adversarial Training



(a) CaffeNet (b) VGG-F (c) VGG-16

(d) VGG-19 (e) GoogLeNet (f) ResNet-152

Figure 11: “Universal perturbations computed for different deep neural network architectures.” [9]

[9] Image taken from “Universal adversarial perturbations” by Moosavi-Dezfooli et al.

Margin Maximization

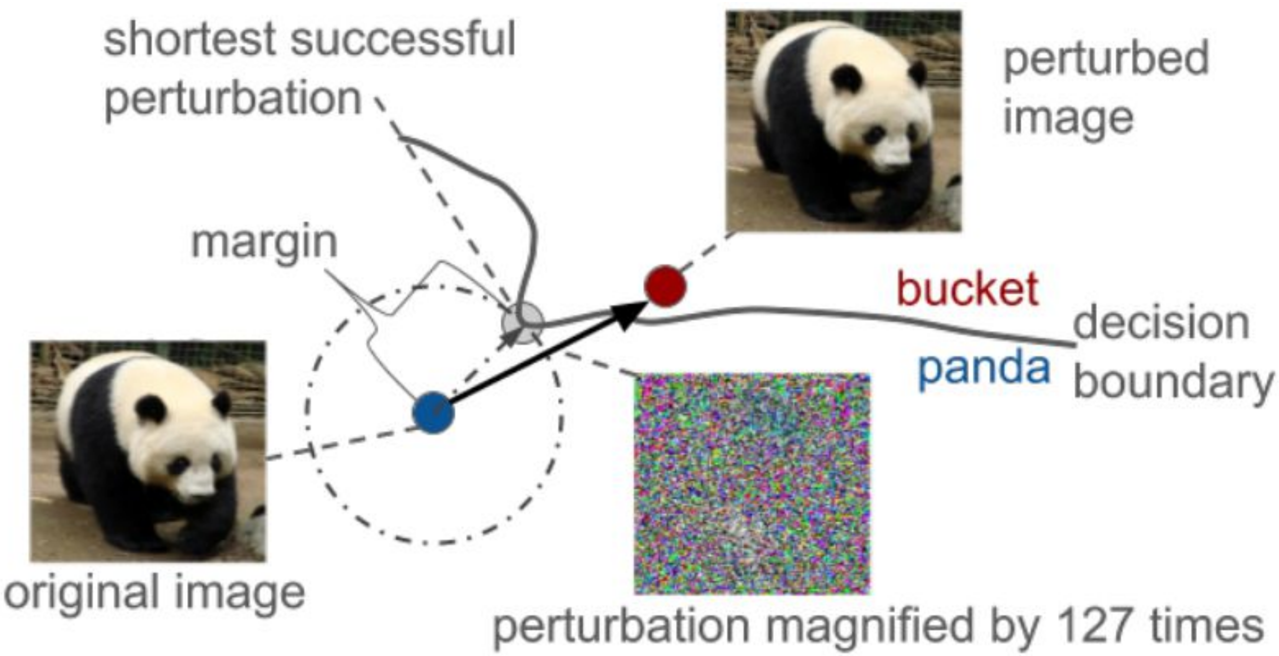


Figure 12: “Illustration of decision boundary, margin, and shortest successful perturbation on application of an adversarial perturbation.” [10]

[10] Image taken from “MMA Training: Direct input space margin maximization through adversarial training” by Ding et al.

Margin Maximization

- maximize margin
- margin = smallest successful perturbation δ^*

$$d_\theta(x,y) = ||\delta^*|| = min||\delta||$$
$$\text{s.t.} \quad \delta : L_\theta^{01}(x+\delta, y) = 1$$

- Two fold problem:

$$\min_\theta \left\{ \sum_{i \in S_\theta^+} \max\{0, d_{max} - d_\theta(x_i, y_i)\} + \beta \sum_{j \in S_\theta^-} l_\theta(x_j, y_j) \right\}$$

Review

Method	Robust accuracy	Training time
K-PGD	baseline	baseline
FGSM	--	+
Free Training	-	+
Fast Training	-	+
Universal Training	--	++
Class-wise universal training	-	+
Margin Maximization	0	0

Open Research Questions

- Precision of finding maximum
- Speed for finding maximum
- Robustness against multiple attack models
- Influence of hyperparameters for robustness

Thank you!

References

- [1] Goodfellow et al., Explaining and Harnessing Adversarial Examples. https://arxiv.org/abs/1412.6572.
- [2] Eykholt et al., Robust physical-world attacks on deep learning visual classification. 2018. https://arxiv.org/pdf/1707.08945.pdf.
- [3] Express, Optical illusion BAFFLES the internet – can YOU spot the straight parallel blue lines?, https://www.express.co.uk/life-style/life/944779/optical-illusions-illusion-pictures-best-viral-puzzle-blue-lines-picture.
- [4] Aleksander Madry et al., Towards Deep Learning Models Resistant to Adversarial Attacks. 2019.

References

- [5] Goodfellow. Adversarial Examples and Adversarial Training. 2016, https://berkeley-deep-learning.github.io/cs294-dl-f16/slides/2016_10_5_CS294-131.pdf.
- [6] Medium. Ignorance is Bliss: Adversarial Robustness by Design with LightOn OPUs. 2020, https://medium.com/@LightOnIO/ignorance-is-bliss-adversarial-robustness-by-design-with-lighton-opus-4f143fa629b
- [7] PyImageSearch. Targeted adversarial attacks with Keras and TensorFlow. https://www.pyimagesearch.com/2020/10/26/targeted-adversarial-attacks-with-keras-and-tensorflow/

References

- [8] Tramer et al. Ensemble adversarial training: Attacks and defenses. 2018.
- [9] Moosavi-Dezfooli et al. Universal adversarial perturbations. 2017. https://arxiv.org/pdf/1610.08401.pdf.
- [10] Ding et al. MMA Training: direct input space margin maximization through adversarial training. 2020. https://arxiv.org/pdf/1812.02637.pdf.