UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

SZEGEDI TUDOMÁNYEGYETEM

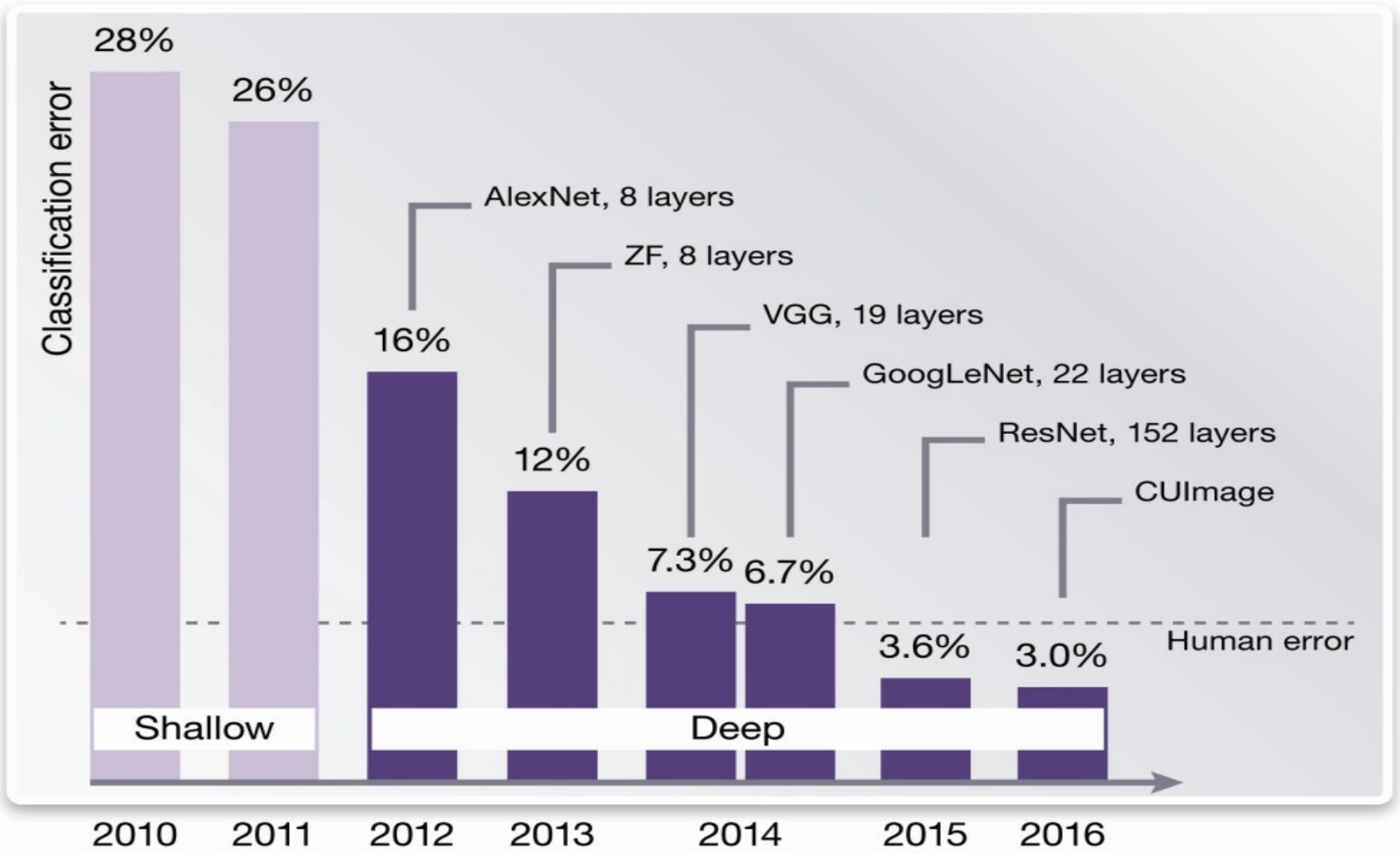
Adversarial Examples in Machine Learning

Jelasity Márk

University of Szeged

AI perception is really good now!

Image recognition over the ImageNet data set

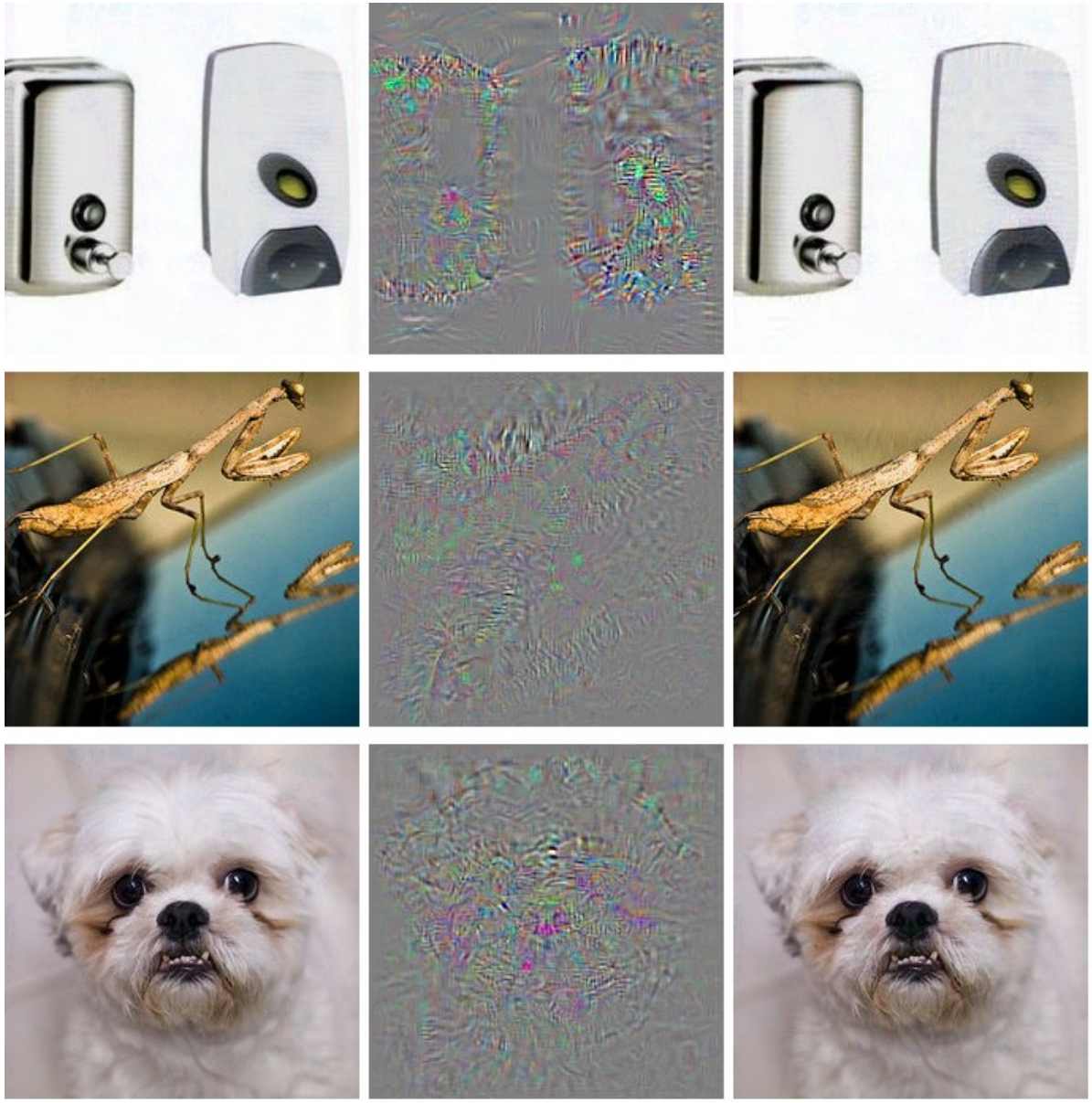


https://semiengineering.com/wp-content/uploads/2017/11/SE_Nov_Article_-figure-1.jpg

Or is It ???

Right column according to neural network is all

Ostriches!



Intriguing properties of neural networks

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus

Some more examples

Milla Jovovich

Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition
Sharif et al

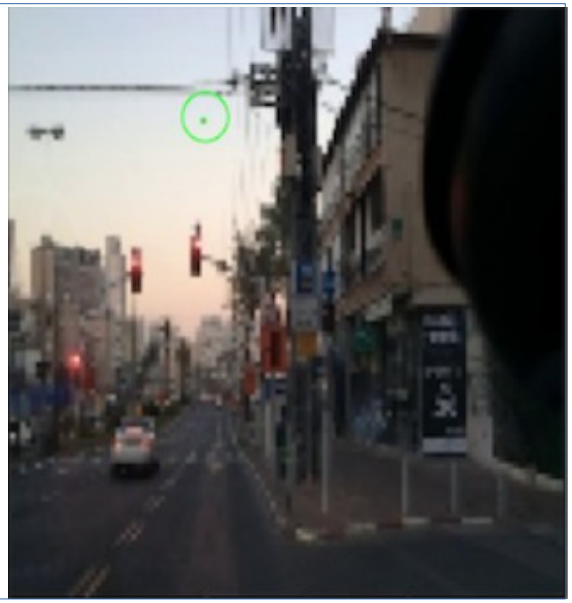
Wearable accessories (top left)
Stickers (top right)
One pixel modified (bottom right)

45 mph speed limit

Robust Physical-World Attacks on Deep Learning Visual Classification
Eykholt et al

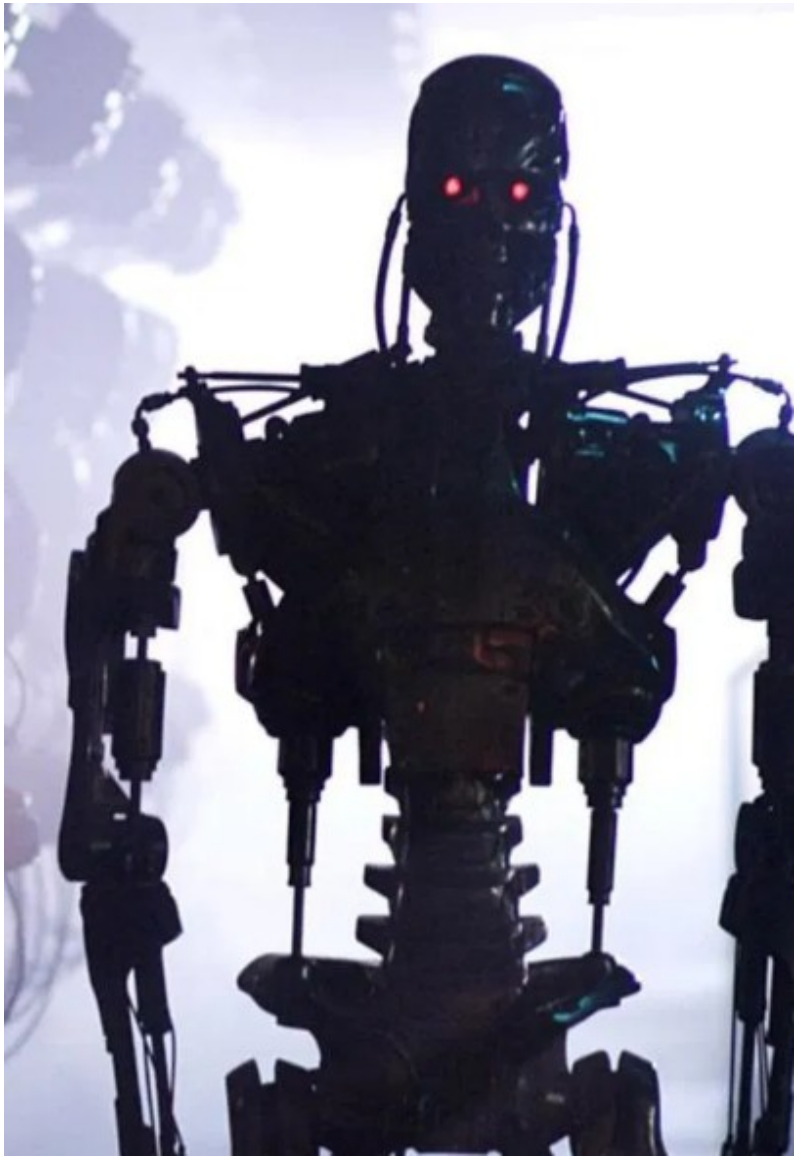
Green light

Feature-Guided Black-Box Safety Testing of Deep Neural Networks
Wicker et al



Security implications

- Robustness of intelligent control systems
  - Self-driving vehicles
  - Industry 4.0 systems
  - Smart-city infrastructure
  - Autonomous weapon systems!
- Bypassing defense solutions
  - Biometric identification
  - Intrusion detection



But the real motivation...

- Some researchers argue that
  - Lots of threats are easier to implement…
  - And so adversarial examples are only of theoretical interest

- Maybe, but still
  - Adversarial examples indicate a shockingly bad understanding of current technology
  - The key problem is the clearly non-human behavior
  - Solving this would solve AI! (will elaborate on this later)

Justin Gilmer, Ryan P. Adams, Ian Goodfellow, David Andersen, and George E. Dahl. Motivating the rules of the game for adversarial example research. Technical Report 1807.06732, arxiv.org, 2018

Our focus

- We focus on the invisible changes
  - very small noise, undetectable to human perception
- There are other issues that might stem from very similar roots
  - Examples when the change is visible and even large (t-shirt patterns, glasses, stickers, etc)
  - Or when there is an image outside the training set distribution (rainy test example, sunny training set)
  - Chances are that solving the “invisible” case will help these too

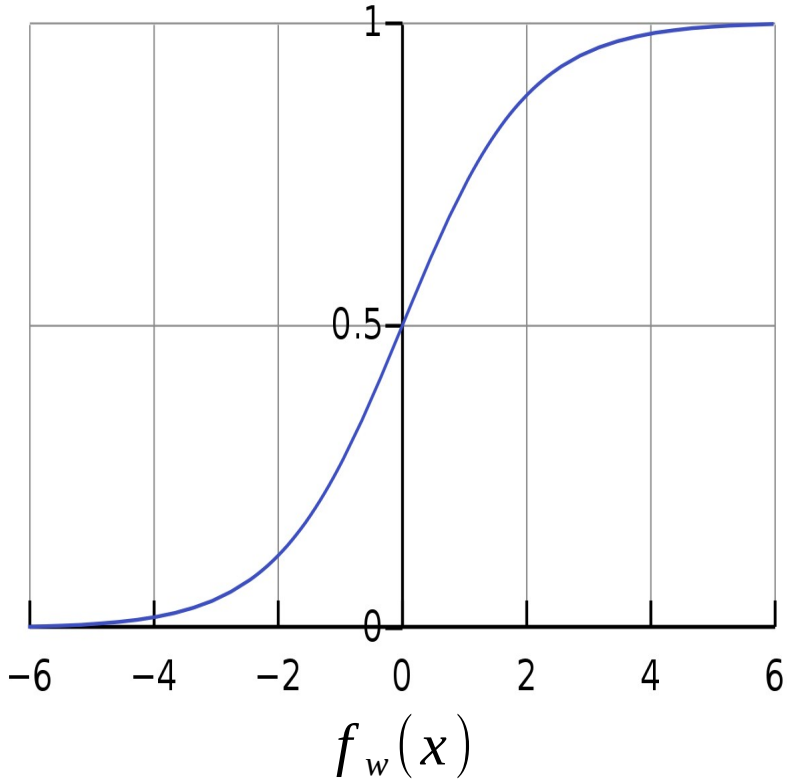
Basic notions of binary classification

- We are given a set of examples (x_i,y_i), where y_i is the class of x_i where y_i is 0 or 1.
- We want a model h(), such that for all i, $h(x_i)\approx y_i$
- h() is often a parameterized function $h_w()$, and the classification problem becomes an error minimization problem in w
- The error is often defined as a sum of the losses of the examples

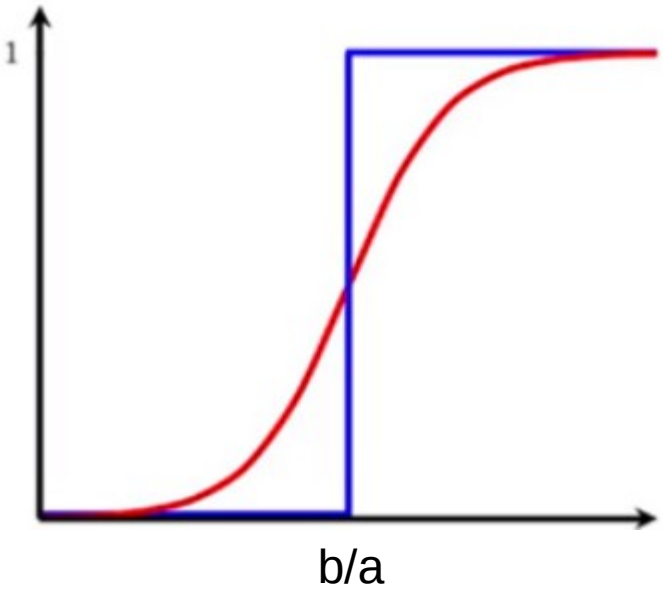
Sigmoid output

- $h_w(x)$ is most often looked for in the form of:
  - We have $f_w(x)$, a non-linear continuous function
  - Then the sigmoid function is applied, the output of which is often interpreted as probabilities
- Other functions are sometimes used too such as tanh, etc.
- The decision boundary is where $f_w(x)=0$

$$\frac{1}{1+e^{-(f_w(x))}}$$



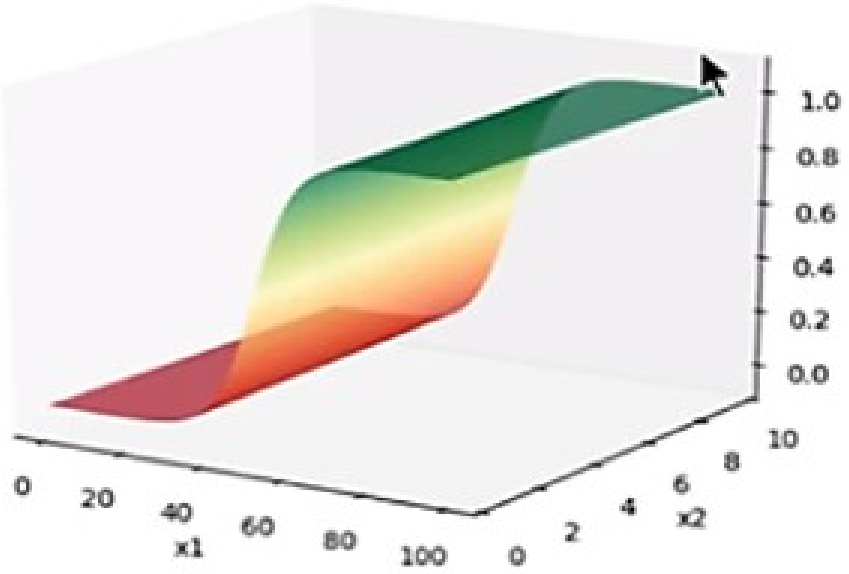
Linear model f_w



1D

$$\frac{1}{1+e^{-(ax+b)}}$$

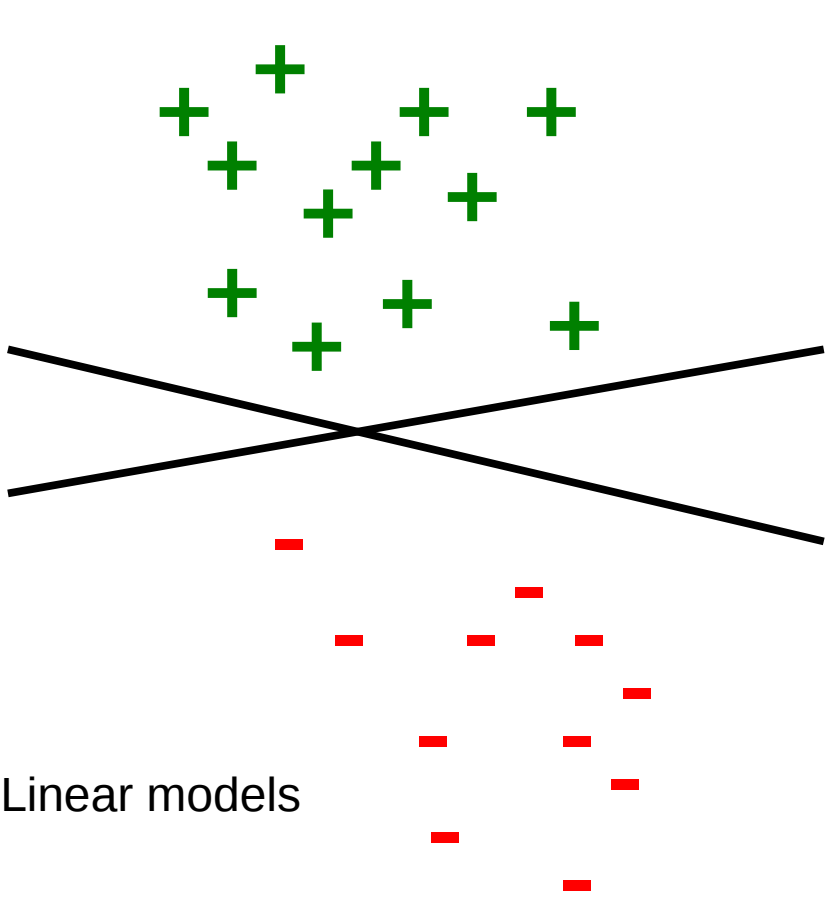
2D



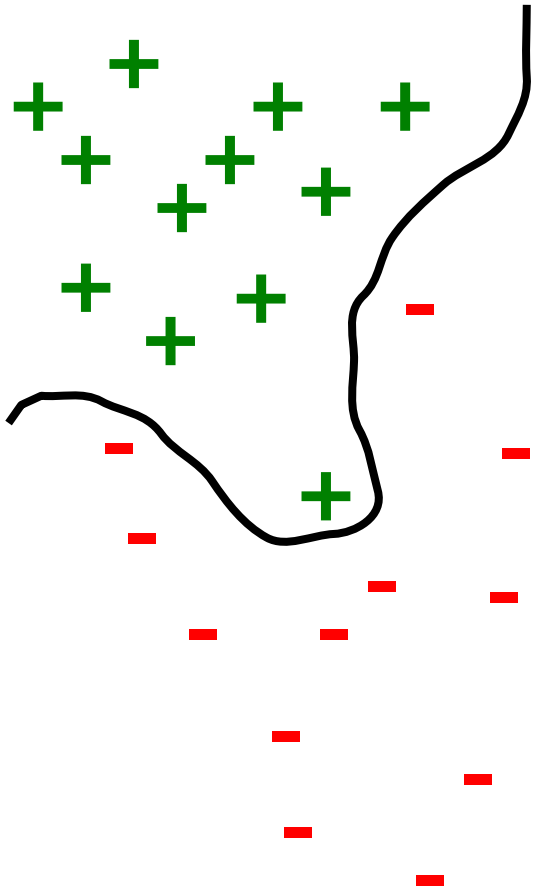
- If $f_{a,b}(x)=ax+b$ is linear then
  - Parameter a represents steepness
  - The bias b defines the decision boundary

Image: https://towardsdatascience.com/sigmoid-neuron-deep-neural-networks-a4cd35b629d7

Decision boundary

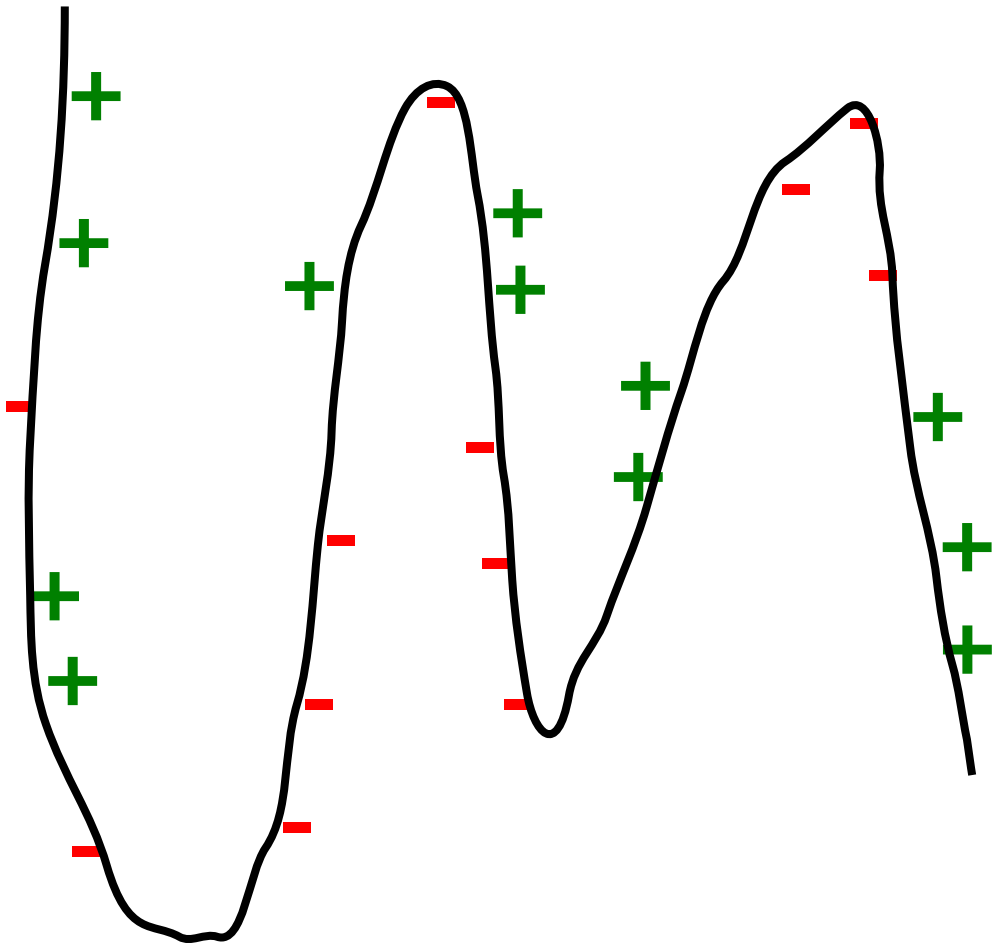


Linear models



Non-linear model

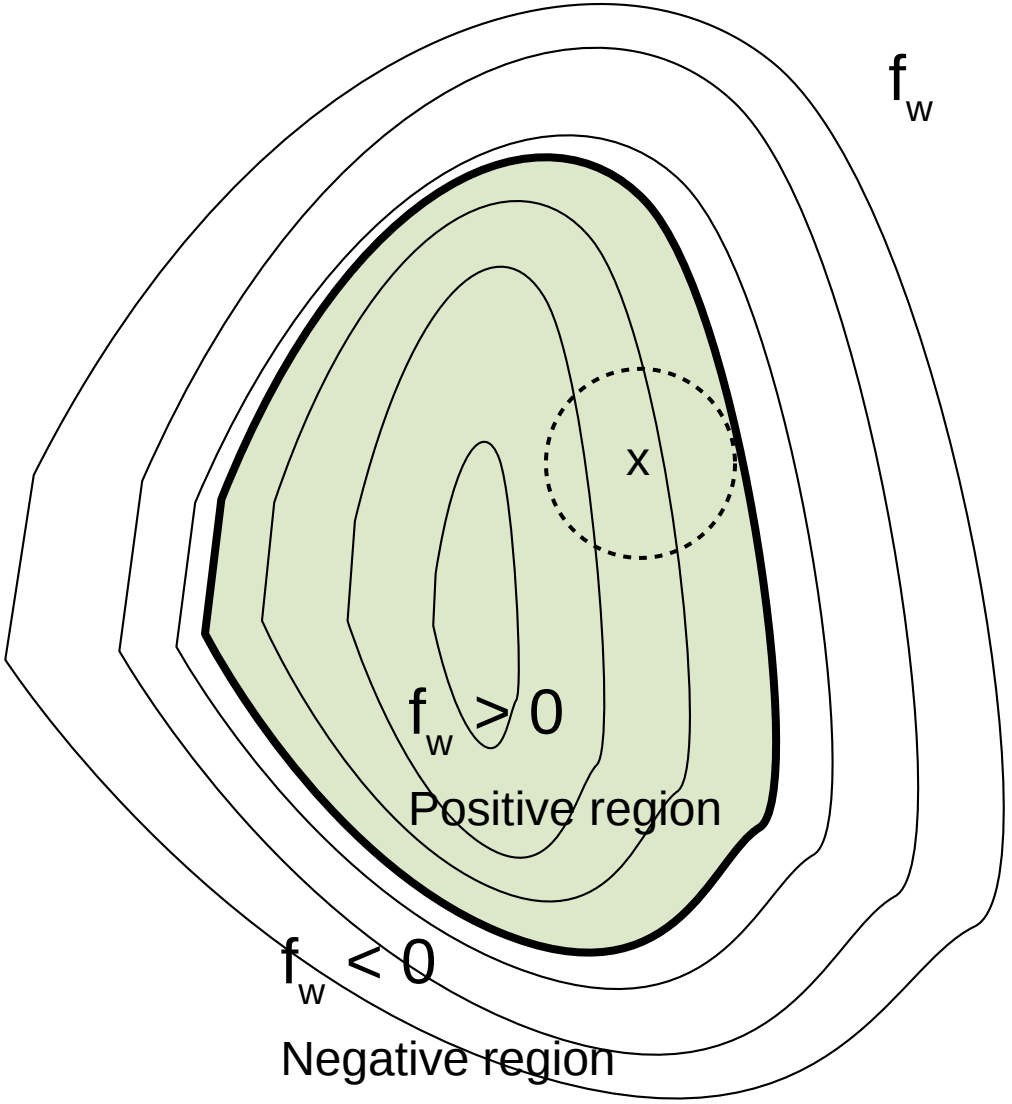
Visualization of adversarial problem



- Distance here represent “perception distance”
  - Normally a proxy is used such as ℓ_2, ℓ_1 or ℓ_∞ distance
- Probably a misleading visualizatoin
  - But food for thought!

The optimization problem

- Finding the smallest adversarial perturbation is a non-linear constrained optimization problem with millions of parameters
  - Illustration of the classification function $f_w(x)$ (without the sigmoid activation), input x, and the minimal perturbation

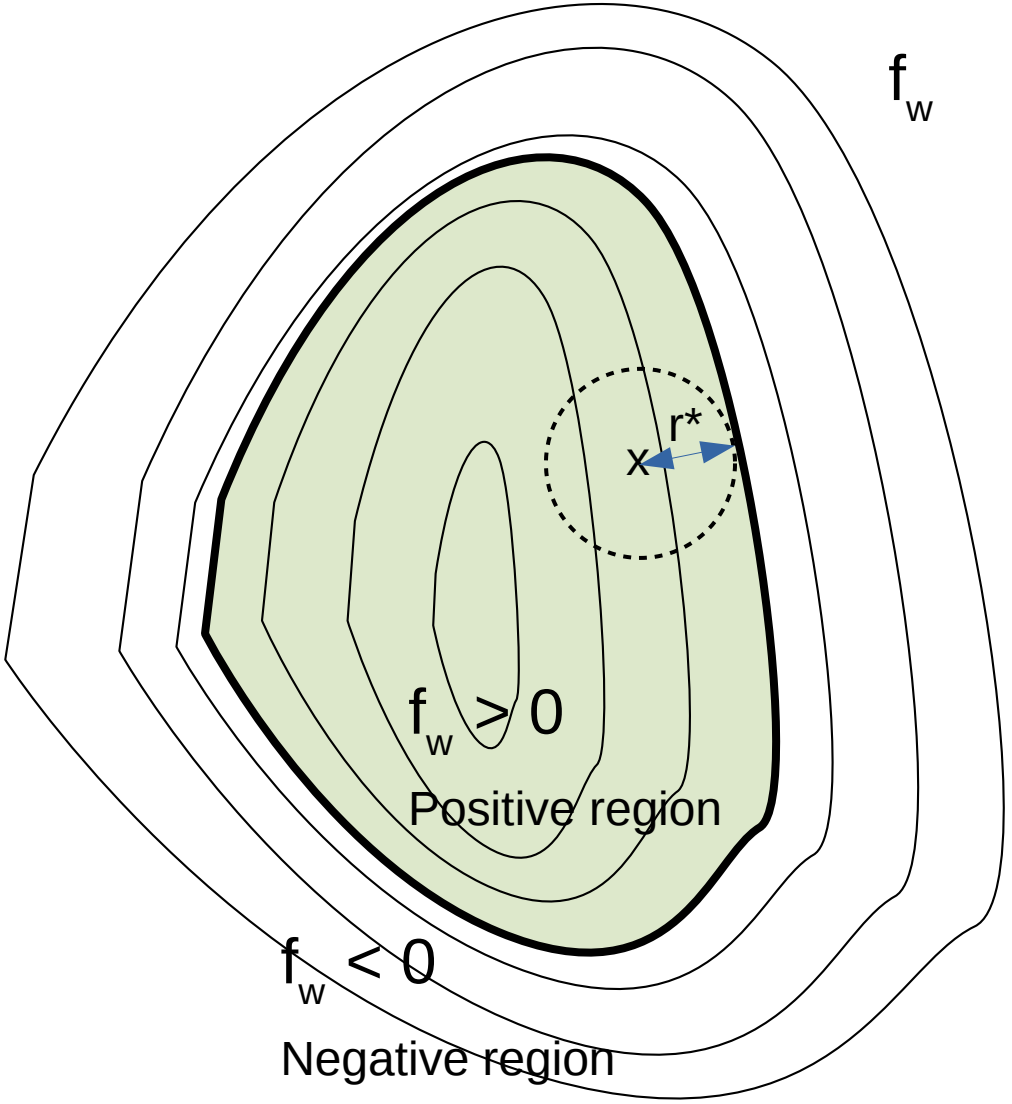


The optimization problem

- Problem formulation
  - Assume an x is given
  - We want to find adversarial perturbation r*

$$r^* = \arg\min_r \|r\|_2 \text{ such that } \operatorname{sign} f_w(x+r) \neq \operatorname{sign} f_w(x)$$

- Other norms are often used
- If input is image, box constraints to the pixels can be added as well

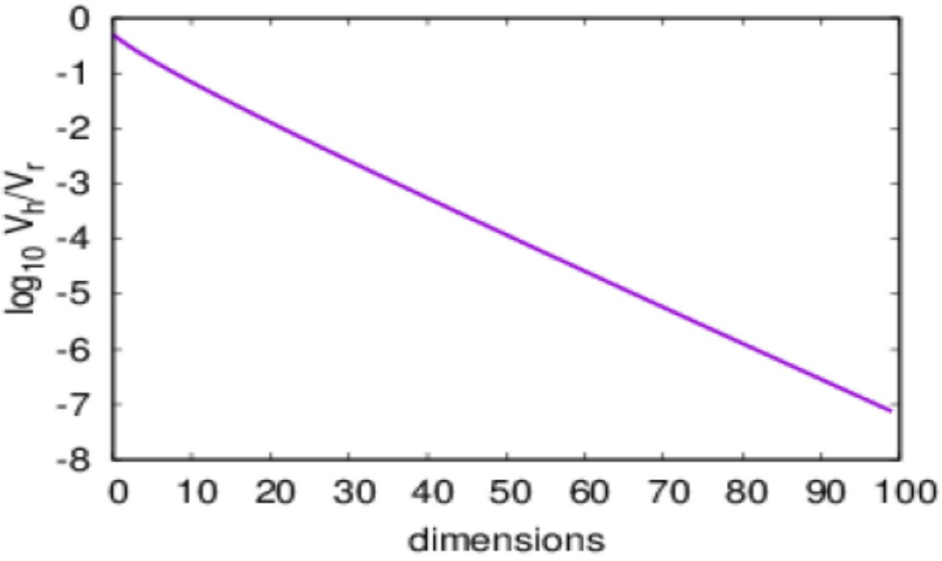


Some areas of research

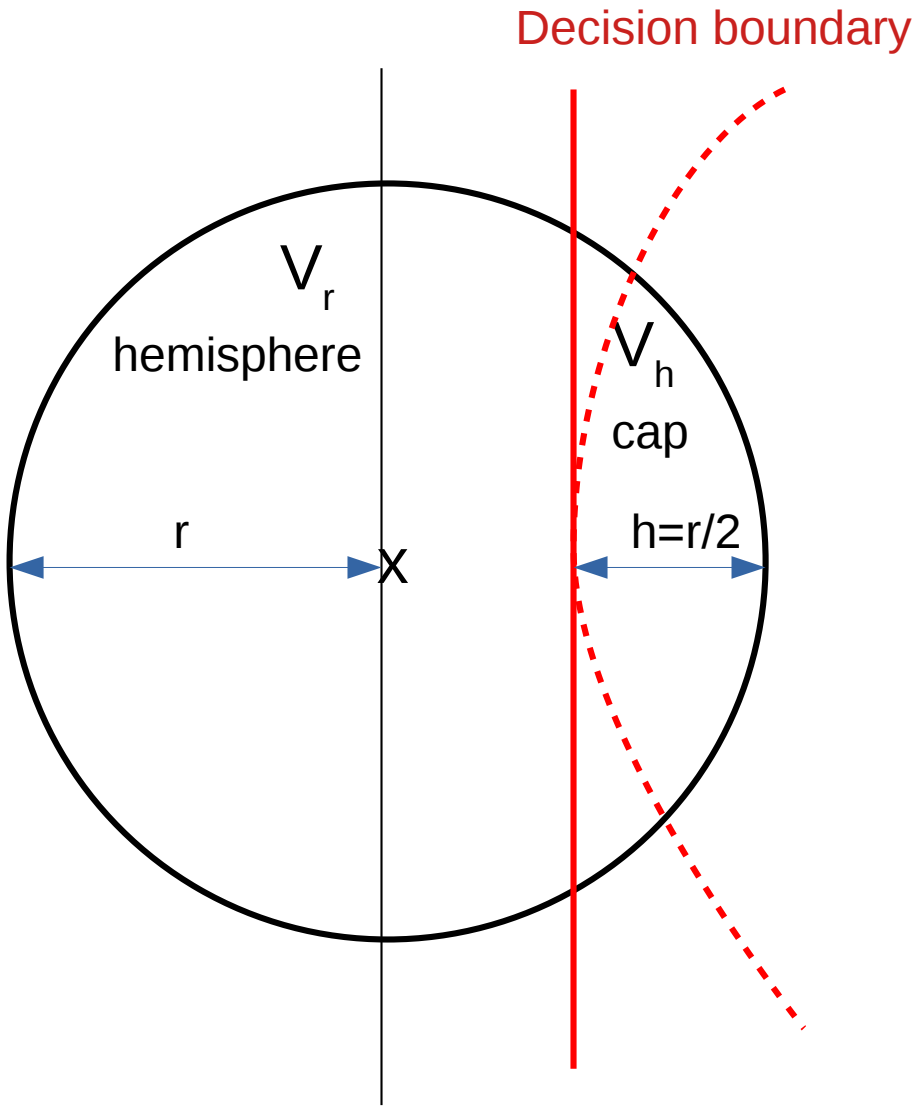
- Attacks
  - Let us find adversarial perturbations for given examples on given models!
- Defenses
  - Let us train models for which the adversarial perturbations are larger in expectation!
- Verification
  - Let us prove mathematically that a given set of examples has at least a given robustness
- Qualitative progress?
  - Let us try to explain the dramatic difference of machine and human perception!

The attack problem is not trivial

- Only a very small volume of examples are adversarial in a ball in high dimensions
  - Random perturbation will be safe most of the time!



- The problem is non-linear, so this small set is hard to find



Attack methods

- Attack methods have been proposed for white-box and black-box settings
- Attacks can involve the physical world or only virtual
- Attacks can be targeted or not, they can involve one or more samples at once, etc
- Most researchers focus on very efficient heuristic methods
  - These can be used in defense methods, we will discuss some of them
- We review the gradient sign method and DeepFool, both white-box attacks

The Fast Gradient Sign Method

- The original formulation:
  - Example x, correct label y, target label t, model parameters θ, fixed perturbation ℓ_∞ length ε
  - $\mathcal{L}$ Is the training loss
- In our notation, in the binary classification case, this is almost the same as:

$$x + \epsilon \text{sign}(\nabla_x \mathcal{L}(\theta, x, y)) \quad \text{untargeted}$$

$$x - \epsilon \text{sign}(\nabla_x \mathcal{L}(\theta, x, t)) \quad \text{targeted}$$

$$x - \epsilon \text{sign}\left(\nabla_x f_w(x)\right) \quad \text{If y=1}$$

$$x + \epsilon \text{sign}\left(\nabla_x f_w(x)\right) \quad \text{If y=0}$$

Ian J. Goodfellow and Jonathon Shlens Christian Szegedy. Explaining and harnessing adversarial examples. In 3rd International Conference on Learning Representations (ICLR), 2015

The Fast Gradient Sign Method

- This has an iterated version as well

$$x_0 = x$$

untargeted

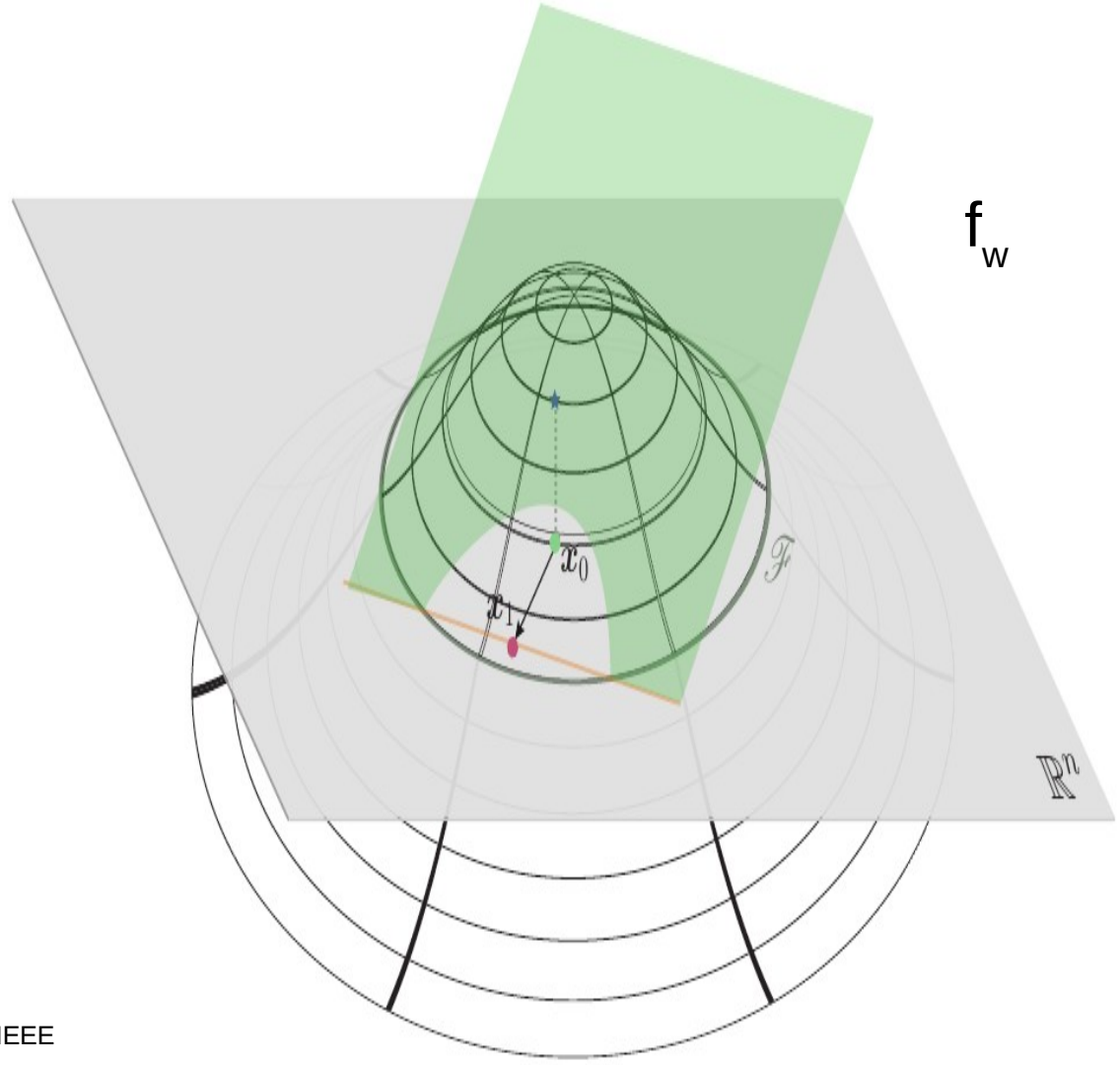
$$x^{t+1} = x^t + \alpha \text{sign}(\nabla_x \mathcal{L}(\theta, x^t, y))$$

  - This version is often called the projected gradient method (PGD)
  - The parameter α and the number of iterations are set so that the final distance is within ε ℓ_∞ distance from x
- It's not clear why they use the sign instead of the (normalized) gradient itself (sign distorts direction and increases ℓ_2 distance)
  - Although pixels could be clipped and rounded each time
- Both FGSM and PGD work within a fixed ε ℓ_∞ ball, so they do not give information about the closest adversarial example (ie, robustness)

Alexey Kurakin, Ian J. Goodfellow, Samy Bengio
Adversarial Machine Learning at Scale. ICLR (Poster)2017

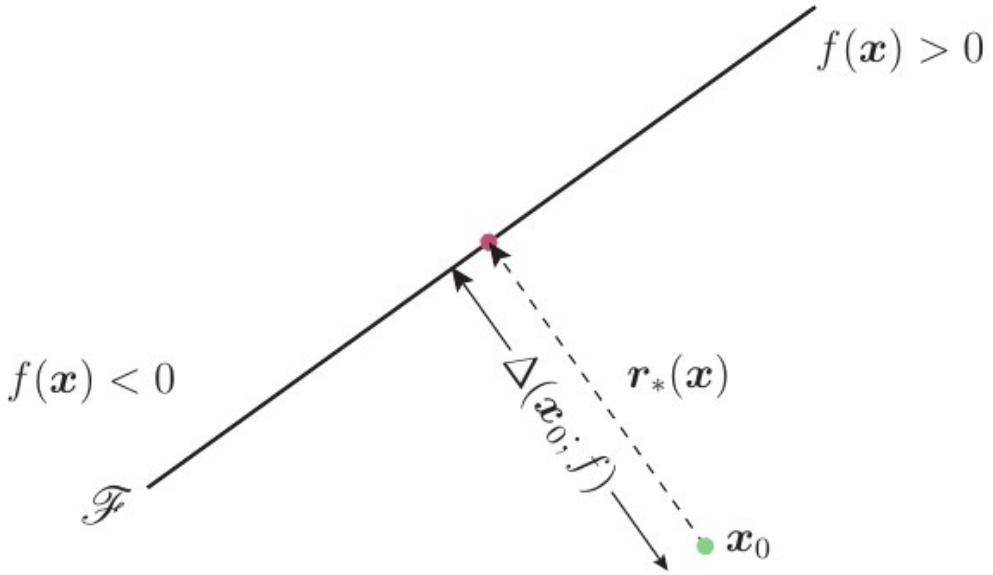
DeepFool

- Here, the goal is to find the smallest perturbation (as definied previously)
- Iterative method based on linear approximations of f_w
  - In other words: Newton's method updates, until we cross the decision boundary
- We kind of hope the point we find will be as close as possible



Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In IEEE CVPR, pages 2574–2582, June 2016

DeepFool



$$r_*(x_0) := \arg\min \|r\|_2$$
$$\text{subject to } \operatorname{sign}(f(x_0 + r)) \neq \operatorname{sign}(f(x_0))$$
$$= -\frac{f(x_0)}{\|w\|_2^2} w.$$

$$r_i \leftarrow -\frac{f(x_i)}{\|\nabla f(x_i)\|_2^2} \nabla f(x_i)$$

- In the linear case
$$f(x) = w^T x + b$$
- the minimal perturbation is the distance from the decision boundary (closed form)
- In the non-linear case we do one iteration step based on the linear approximation of f_w
- The final perturbation is multiplied by 1+η (to move beyond the decision boundary)

A few words on black box attacks

- We can use a substitute model
  - Obtain a model that is very similar to the black box model, eg. by using the black box model to label examples during training
  - Use adversarial examples obtained by attacking this substitute model
- We can also try to create robust adversarial examples
  - Try to find adversarial examples for, preferably diverse, model ensembles to get examples that transfer well
  - Use these examples to attack the unknown model

Defense methods

- Gradient masking
  - Try to make the model gradients extremely flat or even create a non-differentiable model
  - Not a popular method: adversarial examples still exist only they are somewhat harder to find (one needs gradient-free methods)
- Robust learning
  - Learning methods that somehow utilize adversarial examples to make the model more robust
  - This really aims at generating models that do not have adversarial examples at all
- Detection of adversarial examples
  - Can we somehow characterize adversarial examples?

Stochastic gradient descent

- The error is the sum of the losses
- The gradient is
- The full gradient update rule is
- Taking only one random example at a time is the stochastic gradient update rule
  - A mini-batch update consists of more than one examples

$$Err(w) = \sum_{i=1}^{n} l(w, x_i, y_i)$$

$$\nabla Err(w) = \sum_{i=1}^{n} \nabla_w l(w, x_i, y_i)$$

$$w_{t+1} = w_t - \gamma_t \sum_{i=1}^{n} \nabla_w l(w, x_i, y_i)$$

$$w_{t+1} = w_t - \gamma_t \nabla_w l(w, x_i, y_i)$$

Regularization

- Regularization is a broad term meaning we optimize also for something else beside the loss
- The loss is extended with a regularization term, eg

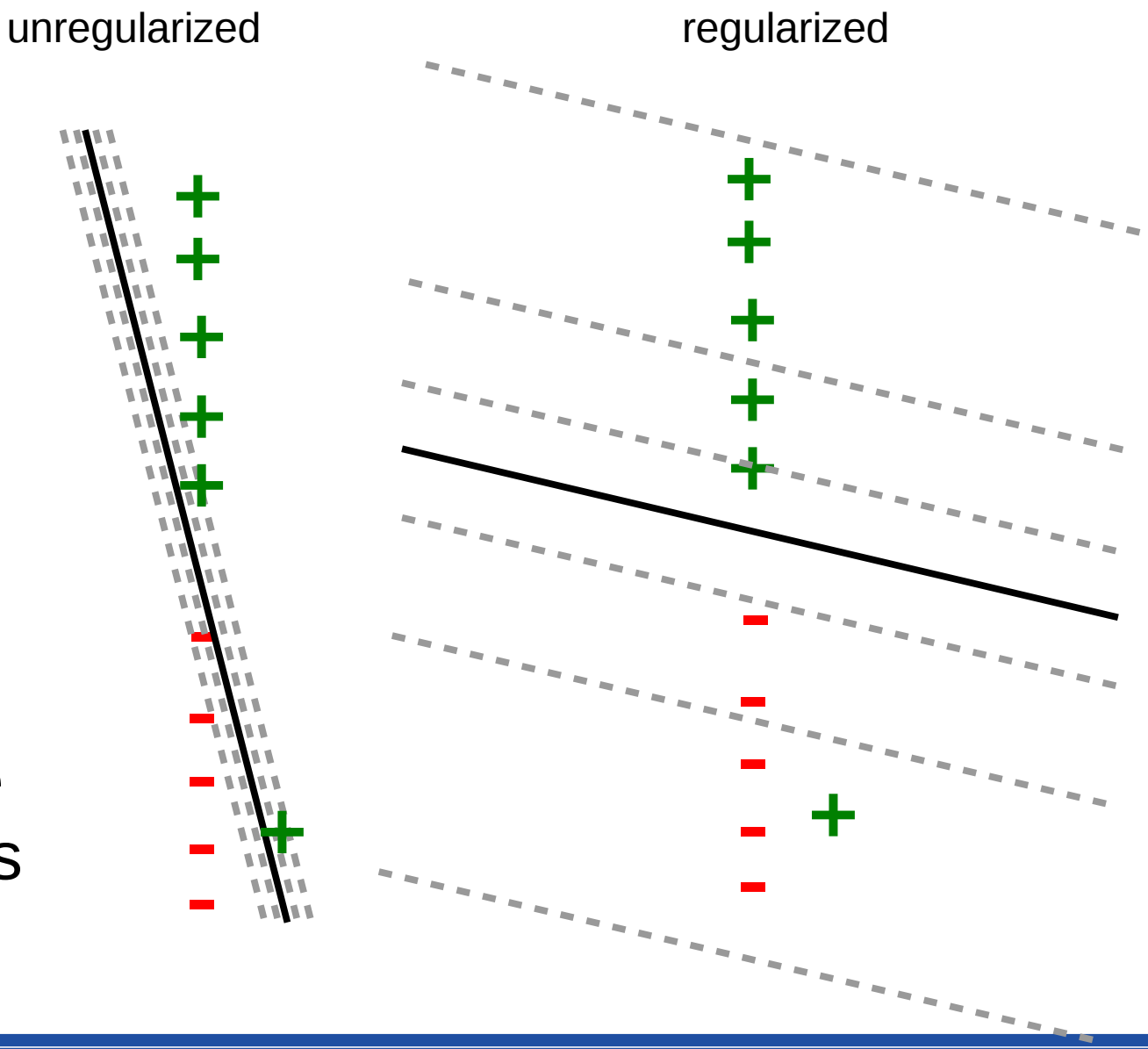
$$l(w, x_i, y_i) + \lambda \|w\|_p$$

  - This is often the length of the parameter vector
- The update rule is then extended with the gradient of this term

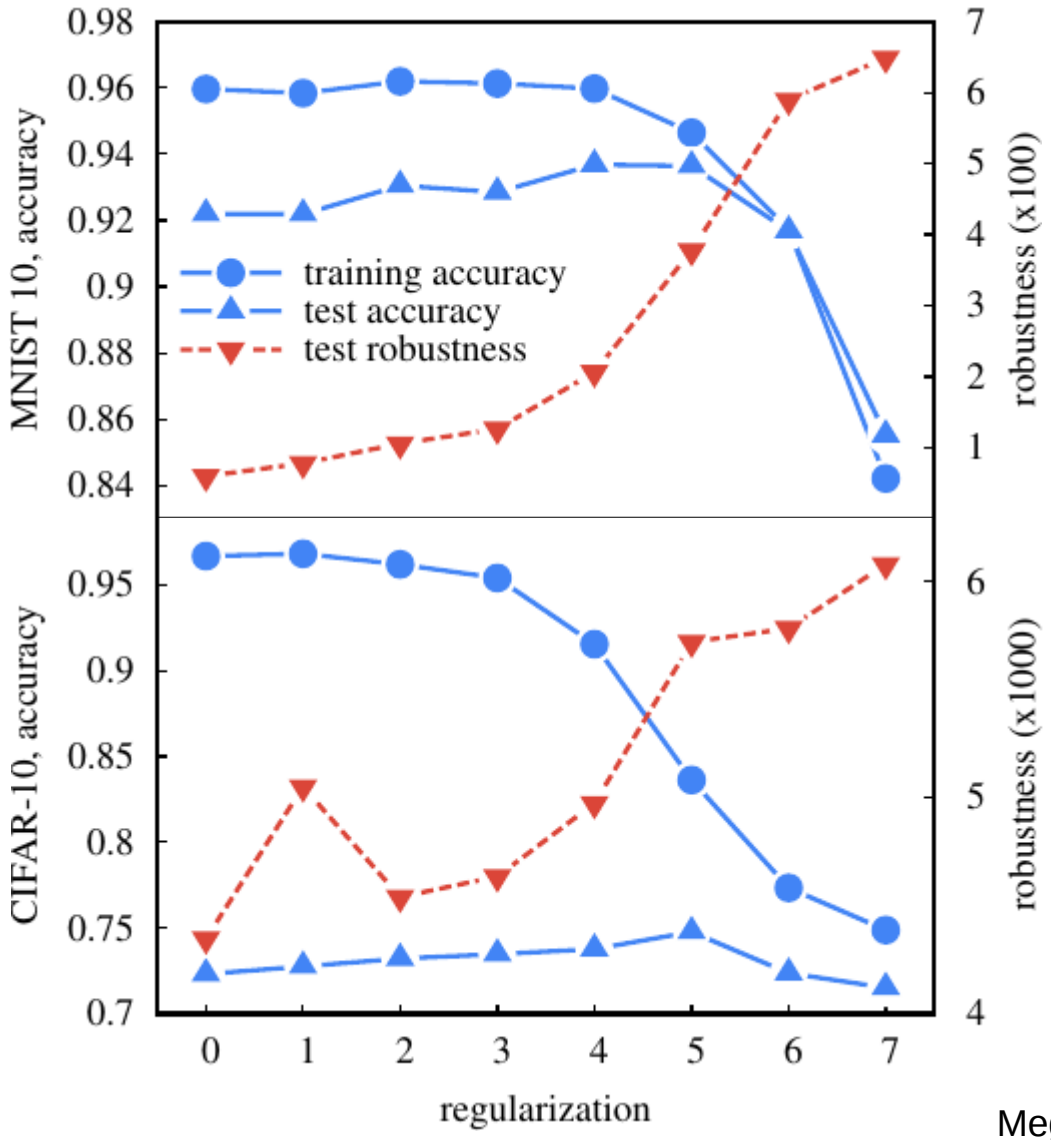
$$w_{t+1} = w_t - \gamma_t \nabla_w l(w, x_i, y_i) - \gamma_t \lambda \nabla_w \|w\|_p$$

Effect of ℓ_2 regularization

- Regularization adds penalty for steepness
- Without this, a few noisy examples can force the decision boundary close to the examples



Accuracy vs. robustness



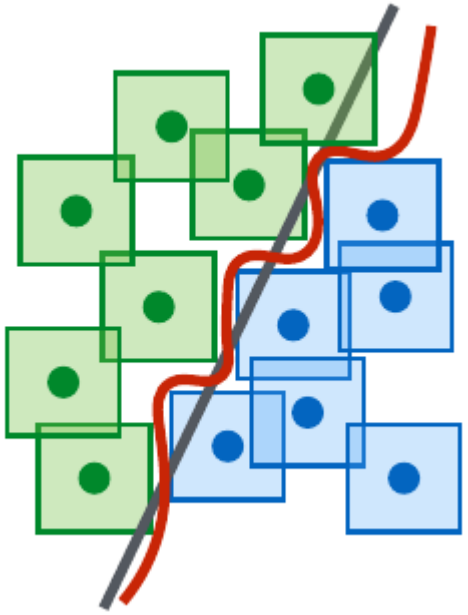
- Robustness often converges slower than accuracy
- Robustness still grows in the over-regularized regime

Megyeri I., Hegedűs I., Jelasity M. submitted

Adversarial training

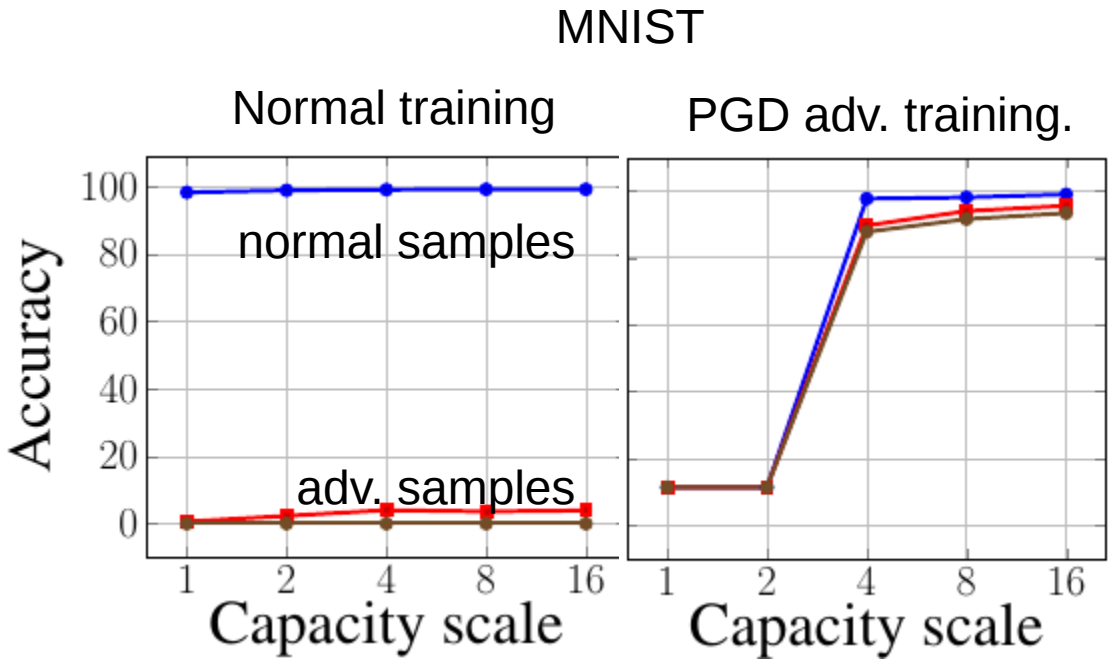
- The idea is to add adversarial examples to the training set
- FGSM-based adversarial training
  - In each mini-batch update, an adversarial example is generated using FGSM for all the examples, and added to the mini-batch
- PGD-based adversarial training
  - In each mini-batch update, all the examples are replaced by their adversarially perturbed versions using PGD
  - This is very good but very expensive as well

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations, 2018



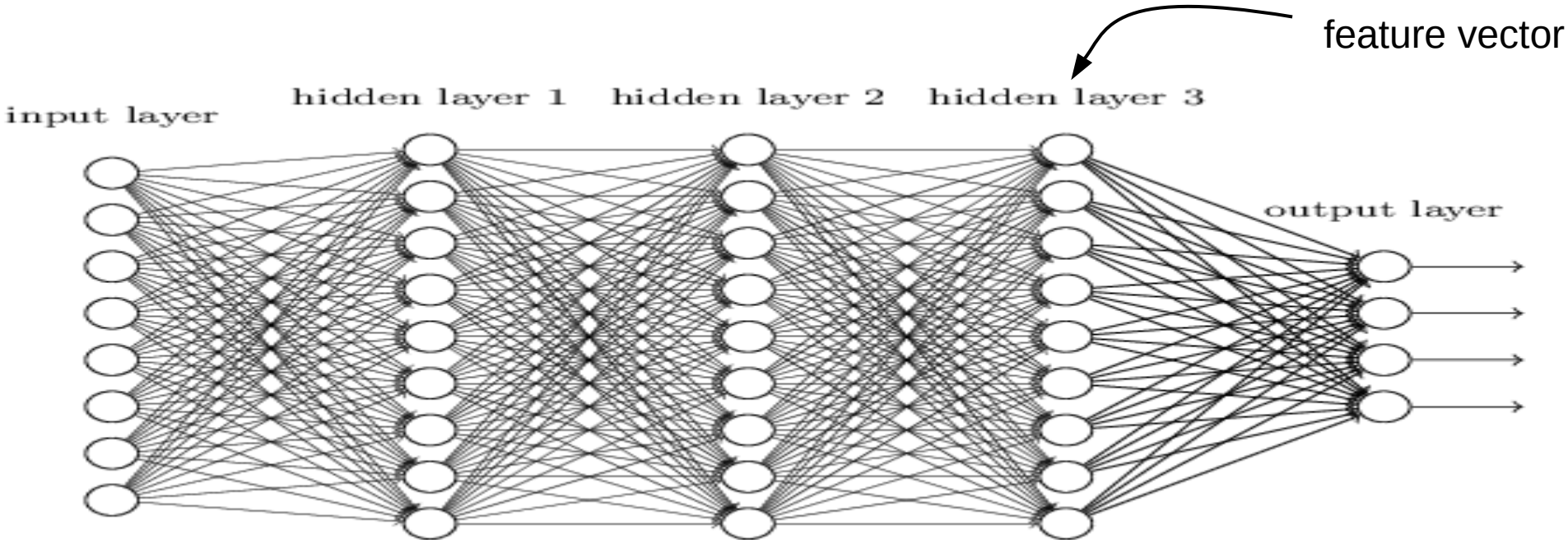
Adversarial training

- Madry et al. make interesting observations about model capacity
  - One needs a model of sufficient capacity so that the increasingly complex decision boundary could be represented



A feature-based look

- The penultimate layer of a feed-forward network can be interpreted as a feature vector, also called representation (R(x))
- The last layer of the network is a linear classifier over this feature vector



http://neuralnetworksanddeeplearning.com/images/tikz41.png

Non-invertible representation

- Using traditional learning, the representation is non-human
  - This can be interpreted as the main reason for adversarial examples
- PGD-based adversarial learning seems to improve this quite a lot!

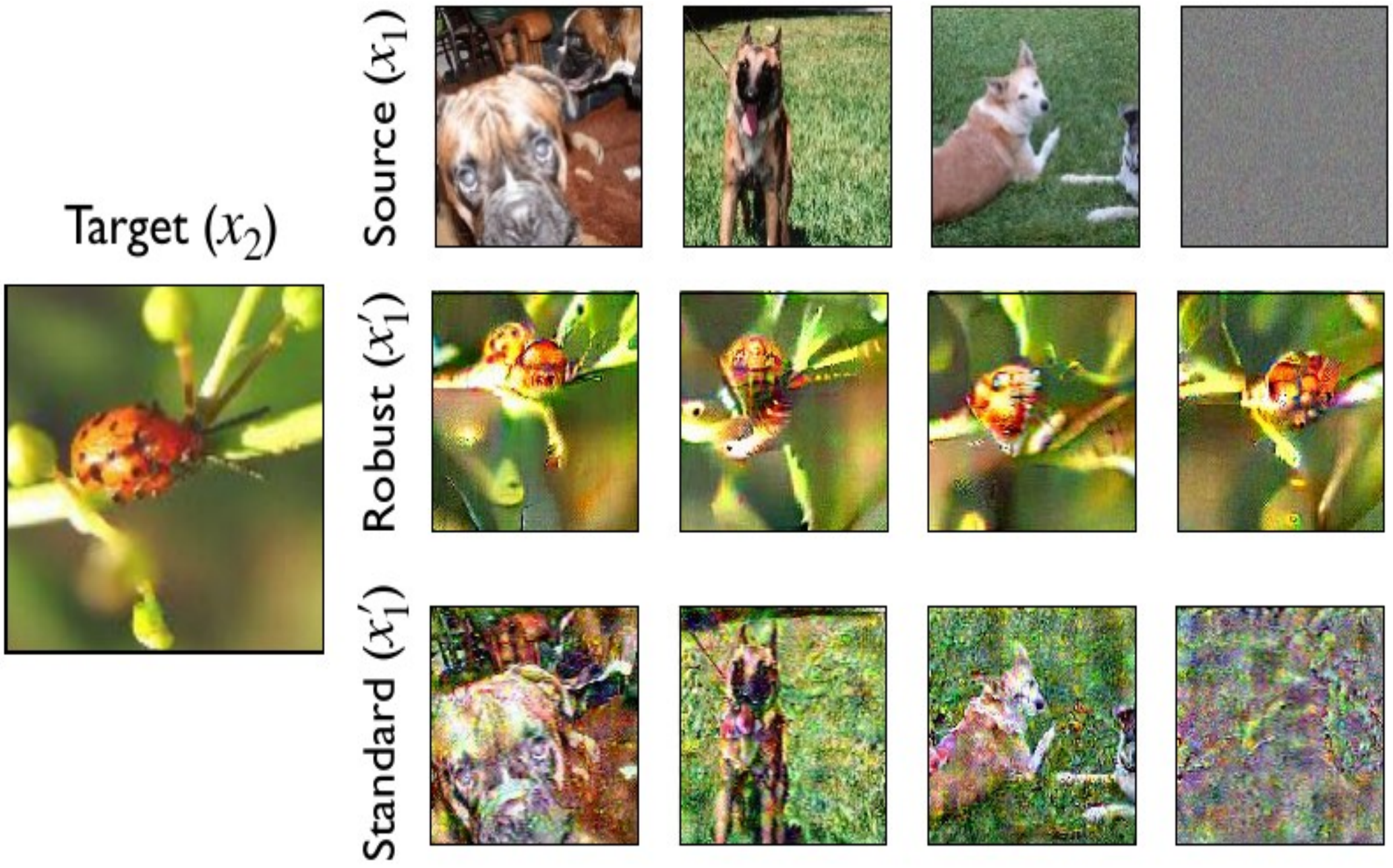
$$x_1' = x_1 + \arg\min_{\delta} \|R(x_1 + \delta) - R(x_2)\|_2$$

$$R(x_1') \approx R(x_2)$$

x_1' x_2

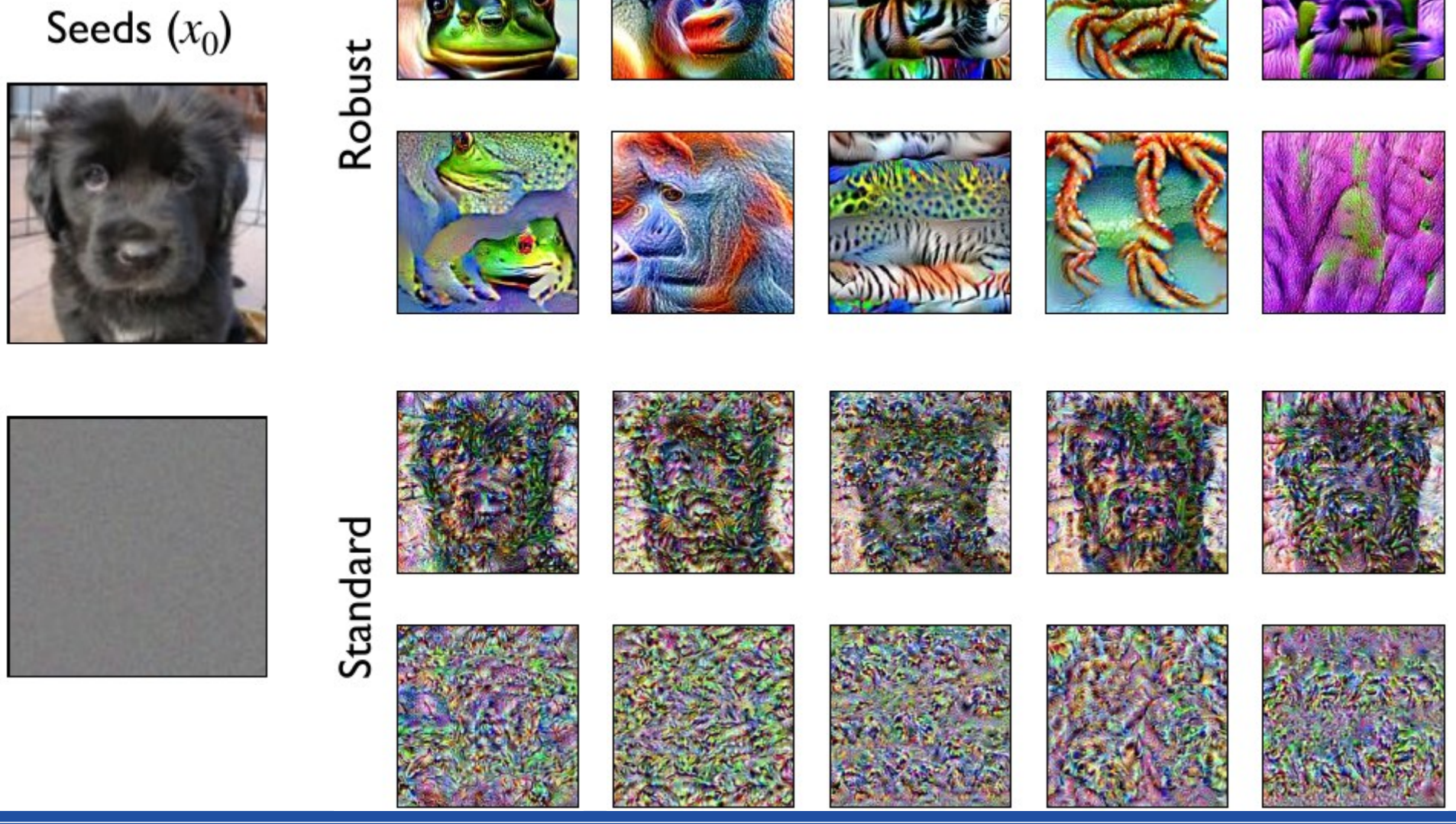
Engstrom et al: Adversarial Robustness as a Prior for Learned Representations.
Arxiv: https://arxiv.org/abs/1906.00945

Invertible representation



Feature visualization

$$x' = \arg\max_{\delta} R(x_0 + \delta)_i$$



Feature visualization

- Lots of visualizations have been published similar to this
- But they apply regularization techniques along with the optimization objective to make the images visually appealing
- Robustly learned networks produce these results without any regularization

Thank you!

Verification: provable results

- There is an arms race between defense and attack methods
- Provable techniques are supposed to give objective characterizations of the robustness of models