UNIVERSITY OF LEICESTER

www.le.ac.uk

Breaking into a Deep Learning Box

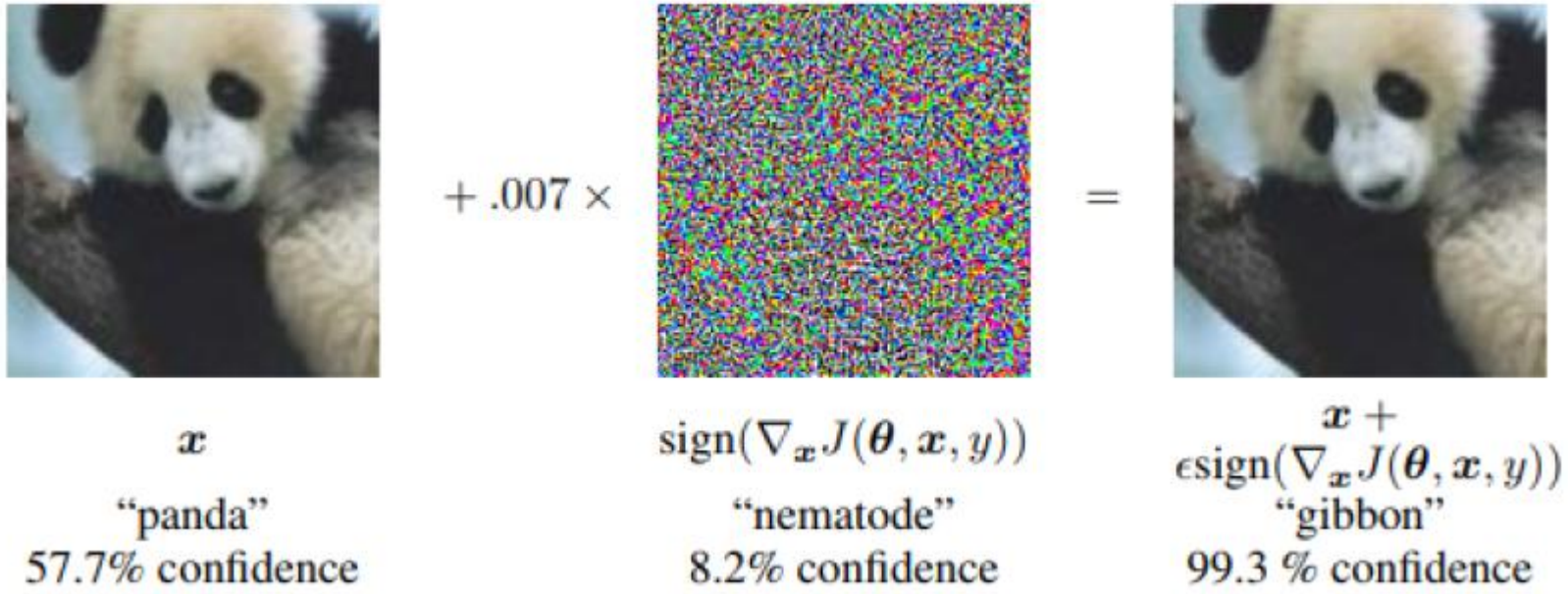
Ivan Y. Tyukin

joint work with Desmond J. Higham, Alexander N. Gorban

ICERM Workshop on Safety and Security in Deep Learning, April 10-11, 2021

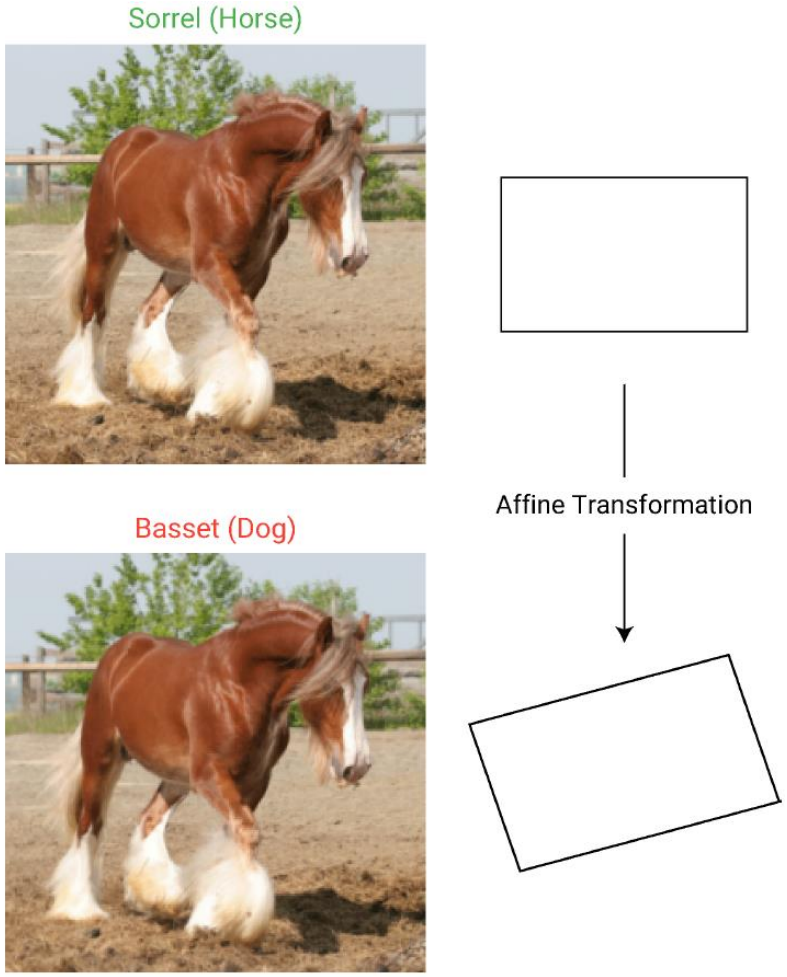
Engineering and Physical Sciences Research Council

Adversarial Examples

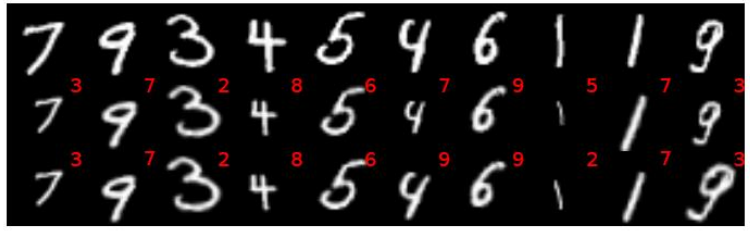


I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in International Conference on Learning Representations, 2015. [Online]. http://arxiv.org/abs/1412.6572

Adversarial Examples



(a) Translations



(b) Similarity transformations

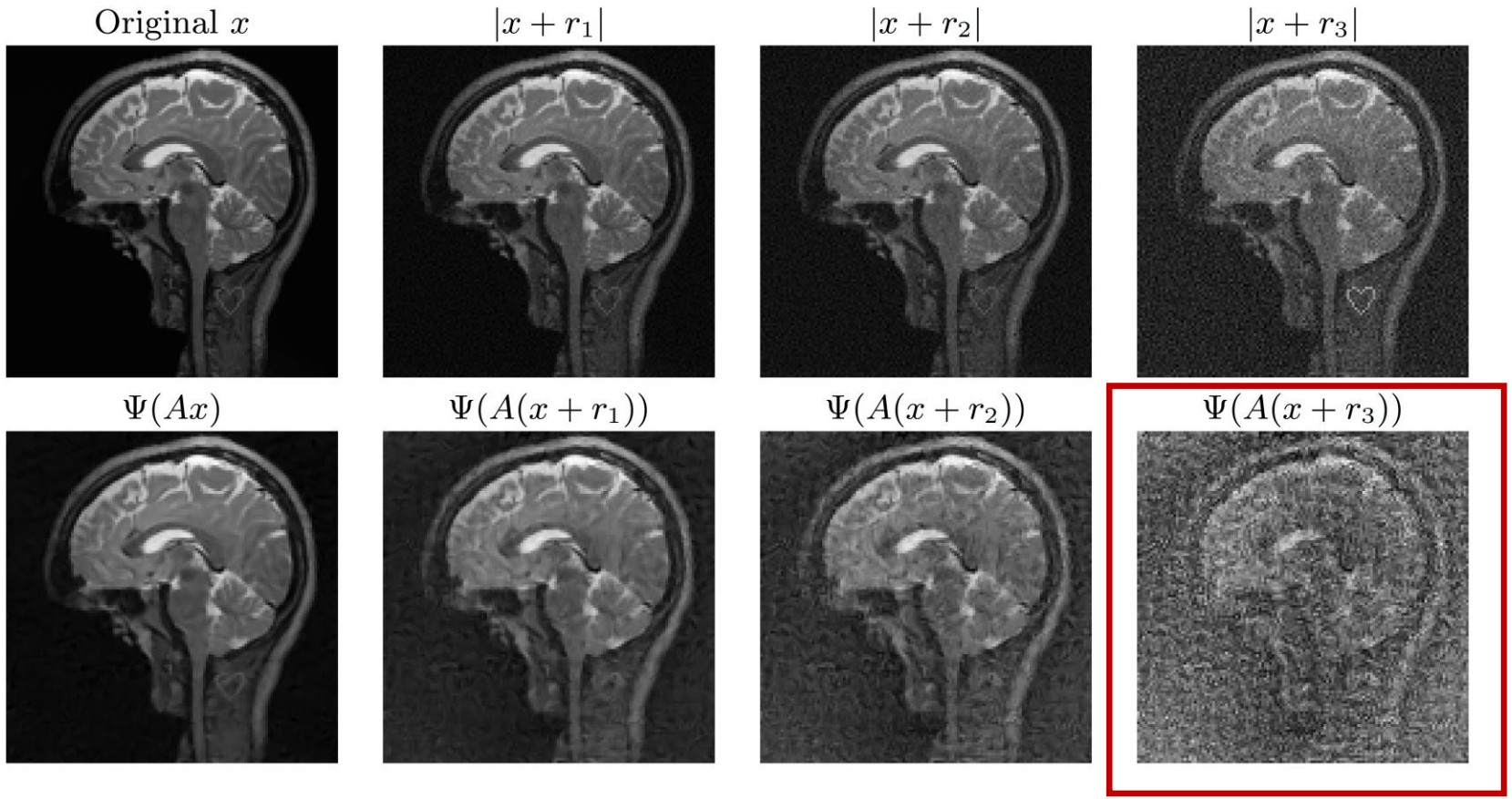
C. Kanbak, S.M. Moosavi-Dezfooli, P. Frossard, “Geometric robustness of deep networks: analysis and improvement”, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. [Online] https://arxiv.org/abs/1711.09115

Adversarial Examples

Subtle Poster	Subtle Poster Right Turn	Camouflage Graffiti	Camouflage Art (LISA-CNN)	Camouflage Art (GTSRB-CNN)
				
				
				

K. Eykholt et al, “Robust physical-world attacks on deep learning visual classification”, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. https://arxiv.org/pdf/1707.08945.pdf

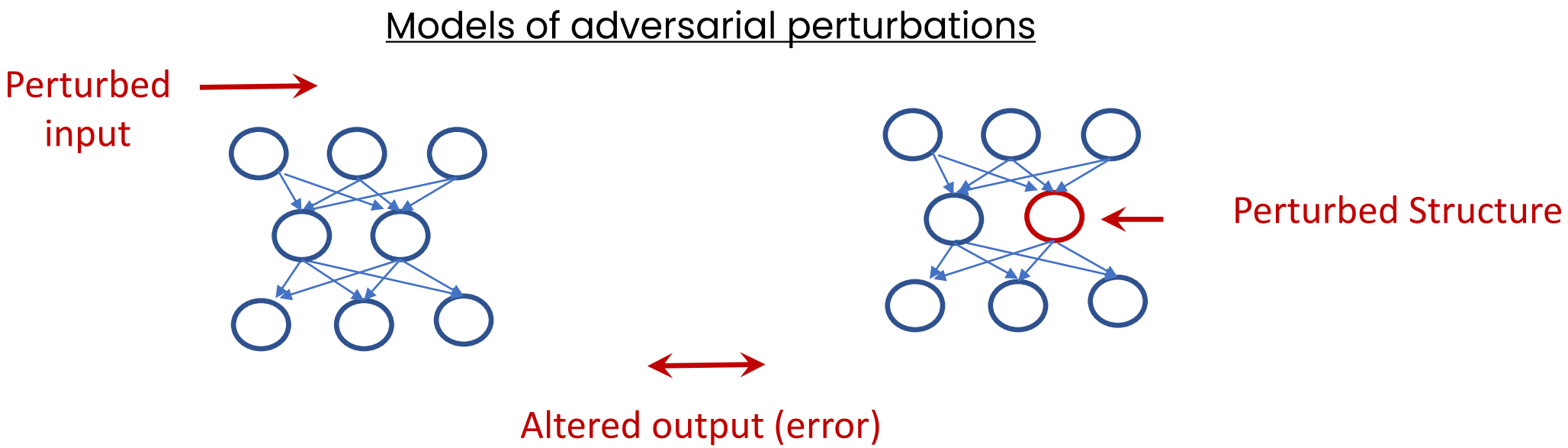
Different from other AI instabilities



N.M. Gottschling, V. Antun, B.Adcock, A. C. Hansen, The troublesome kernel: why deep learning for inverse problems is typically unstable https://arxiv.org/abs/2001.01258

Fundamental Questions

- Why do adversarial examples/attacks exist? Are there fundamental reasons for their occurrence? Can one exploit them (break into our DL models)?



- Is there a unified framework which may help to understand how perturbations to data and models affect data-driven AI/ML systems?

- Can we test for these?

How to test for adversarial or occasional errors, sensitivities and instabilities?

Q1: For a reasonably large training dataset $\{x_1, x_2, \dots, x_M\}$, representing working environment, an appropriate class of perturbations P, a given "tolerance" parameter Δ, and an appropriate metrics $||\cdot||$,

test if the model's output for $x_i + \varepsilon$ (for all $\varepsilon \in P$, $||\varepsilon|| < \Delta$) is consistent with what is expected

A: determine conditions when the model's outputs on perturbed data are different from the model's behaviour on unperturbed data

Formal definition

A *classifier* is a map

$$F\colon\ B_n \to L \subset R,$$

B_n - a unit n – ball in R^n, L – a set of labels

Definition (adversarial example) *For the given classification map F, an element x admits a δ-adversarial example $y(x)$ if*

$$F(x) \neq F\big(y(x)\big) \text{ and } \|x - y(x)\| \leq \delta,\ y(x) \in B_n$$

The ball B_n represents the classifier's "feature space". An element of the feature space can be an image or its representation in a data-driven model (e.g. outputs of hidden layers in feedforward networks)

Theoretical Framework: Adversarial Examples

Features' probability (density) distribution:

$$p: \quad B_n \times L \to R_{\geq 0}$$

Let $A \in L$, then

$$p_A(x) = p(x|l = A), \quad p(x|l = A) = \frac{p(x,A)}{P(A)}, \quad P(A) = \int_{B_n} p(x, A)dx$$

For notational convenience, we denote

$$B_n(r, x) = \{z \in R^n \big| \, \|x - z\| \leq r\}, \quad S_{n-1}(r, x) = \{z \in R^n \big| \|x - z\| = r\}$$

Theoretical Framework: Adversarial Examples

Assumption 1. *There exist a label $A \in L$ and an associated set $C_A \subset B_n$, a number $r_A \in (0,1)$, a vector $x_A \in B_n$, a positive constant C, and $\nu \in (0,1]$ such that*

A1) The set $C_A \subset B_n(r_A, x_A)$

A2) $F(x) = A$ *for all* $x \in C_A$ *and there is a* $\Delta > 0$ *such that for any* $x \in S_{n-1}(r_A, x_A)$ *there is a* $y(x) \in B_n$*:*

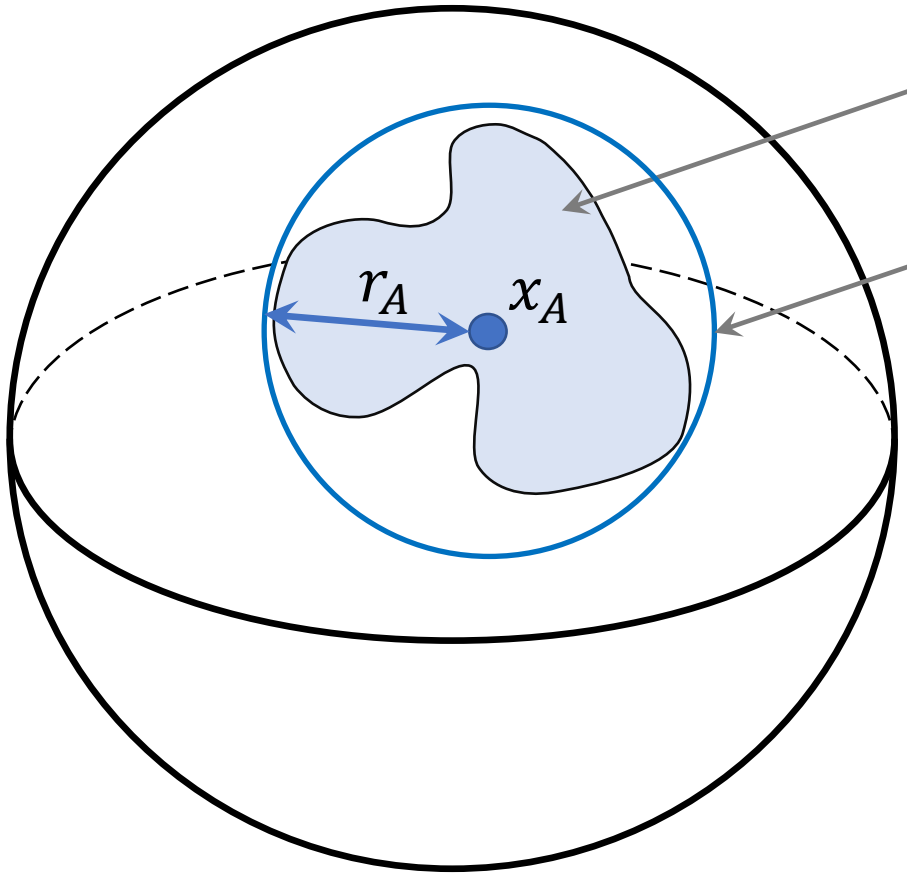
$$F\big(y(x)\big) \neq A, \qquad \|x - y(x)\| < \Delta$$

A3) *The probability density* p_A *satisfies:*

$$p_A(x) \leq \frac{1}{V_n(B_n)} \frac{C}{r_A^n} \quad \forall\, x \in B_n(r_A, x_A)$$

$$\int_{C_A} p_A(x)dx \;\geq \nu$$

Theoretical Framework: Adversarial Examples

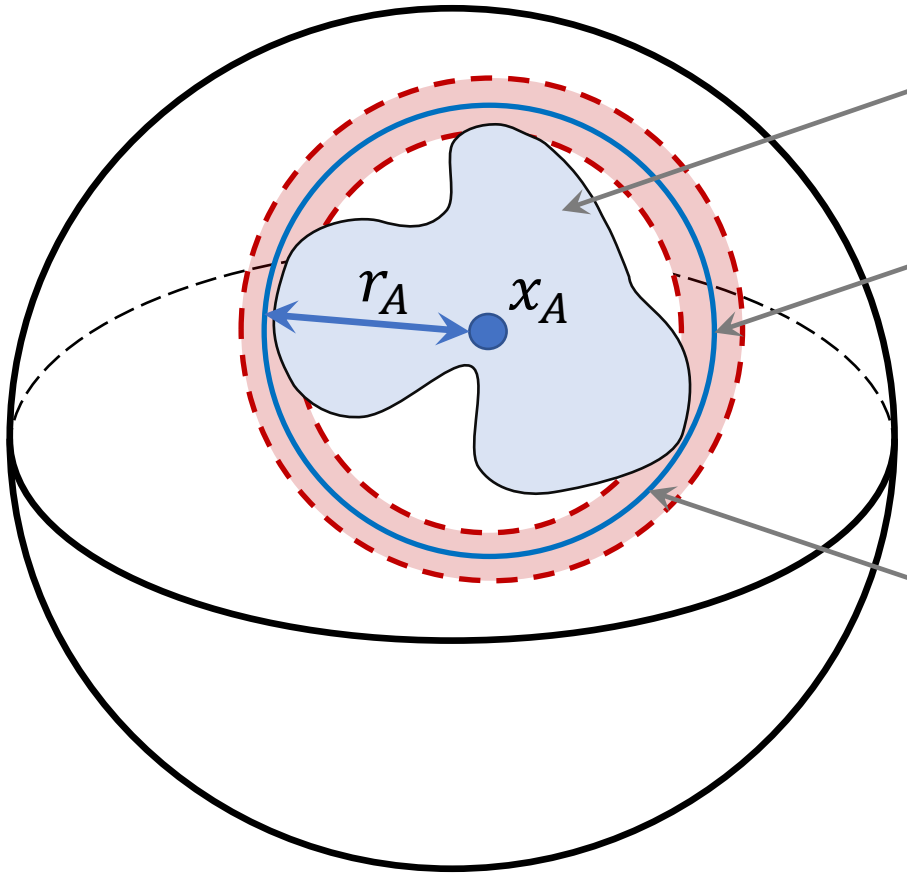


The set C_A

The ball $B_n(r_A, x_A)$

Assumption, Part A1)

Theoretical Framework: Adversarial Examples



The set C_A

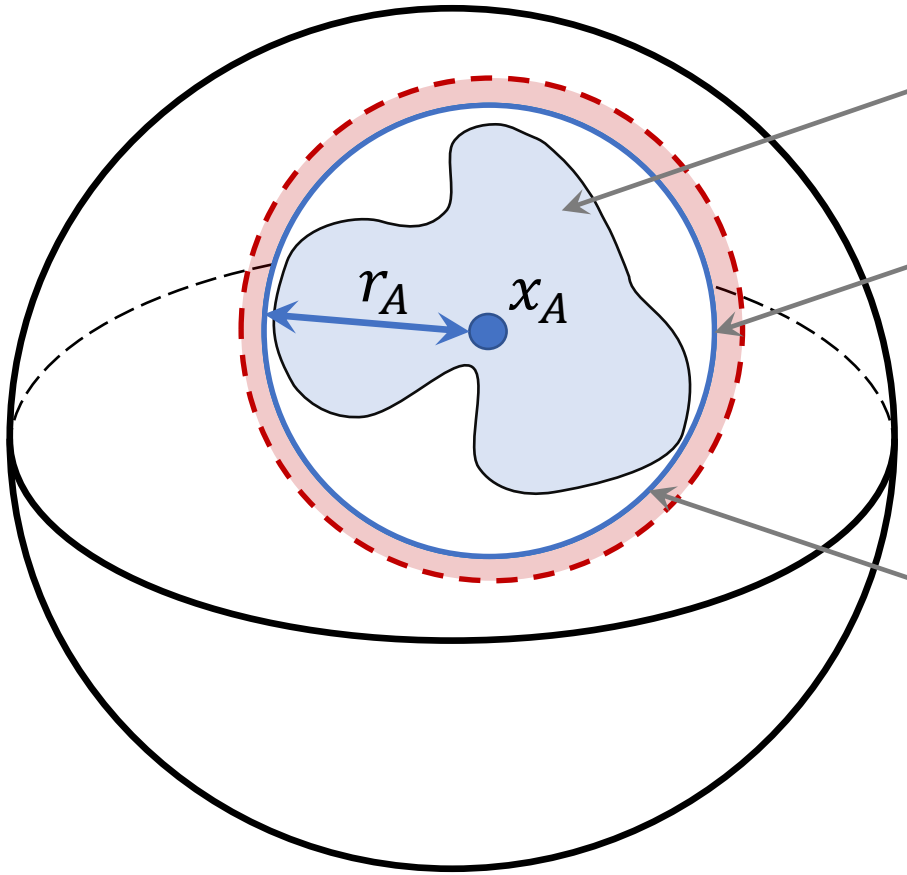
The ball $B_n(r_A, x_A)$

Assumption, Part A2)

The Δ-neighborhood of the ball $B_n(r_A, x_A)$:

for any point x on the boundary of the ball there is an element $y(x)$ within the distance Δ whose class label is not A

Theoretical Framework: Adversarial Examples



The set C_A

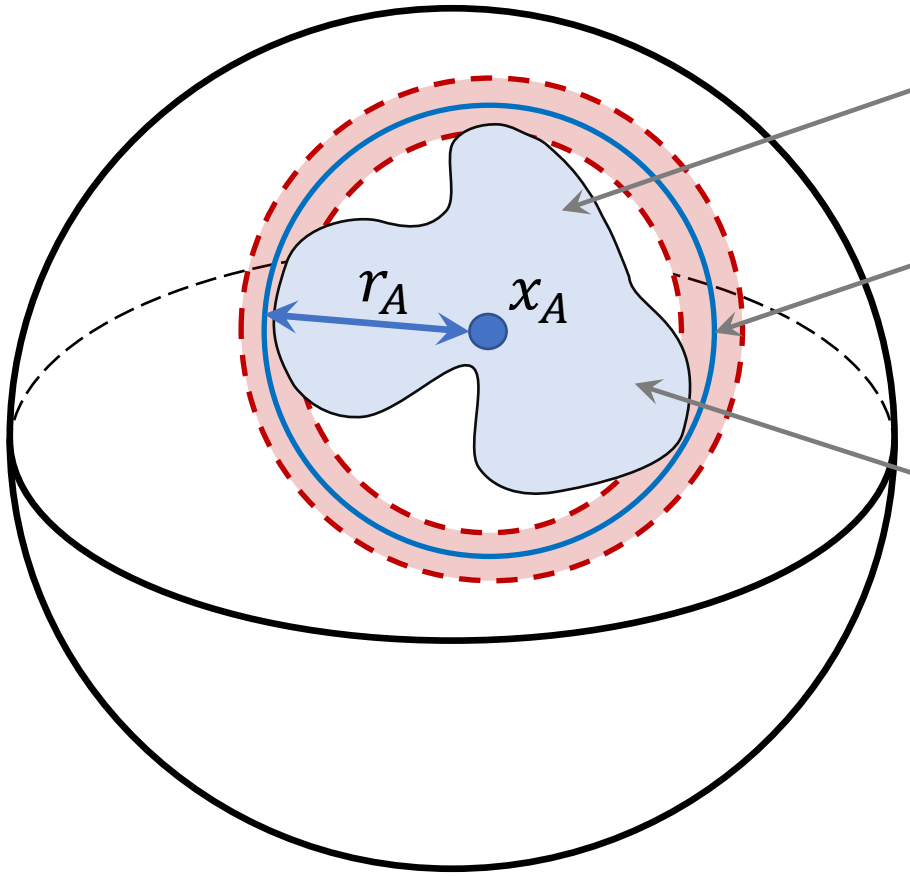
The ball $B_n(r_A, x_A)$

Assumption, Part A2)

The Δ-neighborhood of the ball $B_n(r_A, x_A)$:

for any point x on the boundary of the ball there is an element $y(x)$ within the distance Δ whose class label is not A

Theoretical Framework: Adversarial Examples



The set C_A

The ball $B_n(r_A, x_A)$

Assumption, Part A3)

The measure of the set C_A is at least ν

Theoretical Framework: Adversarial Examples

Theorem 1 (When adversarial examples are typical).

Consider a classification map F and a probability distribution with probability density function p satisfying Assumption 1. Let a sample (x, l) be drawn from this distribution.

Then the probability that x admits a $(\Delta + \varepsilon)$-adversarial example is at least

$$P(A) \max\left\{\nu - C\left(1 - \frac{\varepsilon}{r_A}\right)^n, 0\right\}$$

Theoretical Framework: Adversarial Examples

Corollary 1. *$\Delta + \varepsilon$-adversarial examples are expected to occur (subject to Assumption 1) if the dimension of the classifier's feature space is high enough*

$$n > (\log \nu - \log C\,) \left[\log\left(1 - \frac{\varepsilon}{r_A}\right)\right]^{-1}$$

Remark 1. *An exponential probability bound can be derived*

$$P(A) \max\left\{\nu - C \exp\left(-\frac{\varepsilon\, n}{r_A}\right), 0\right\}$$

Theoretical Framework: Adversarial Examples

Remark 2. *An alternative approach to look at adversarial examples can be established within the framework of isoperimetric inequalities (M. Gromov). This framework enables to link the probability of adversarial examples with dimension (e.g. Shafahi et al., 2019)*

$$n \sim O(1/\varepsilon^2) \quad (n \sim 10{,}000 \text{ if } \varepsilon = 0.01)$$

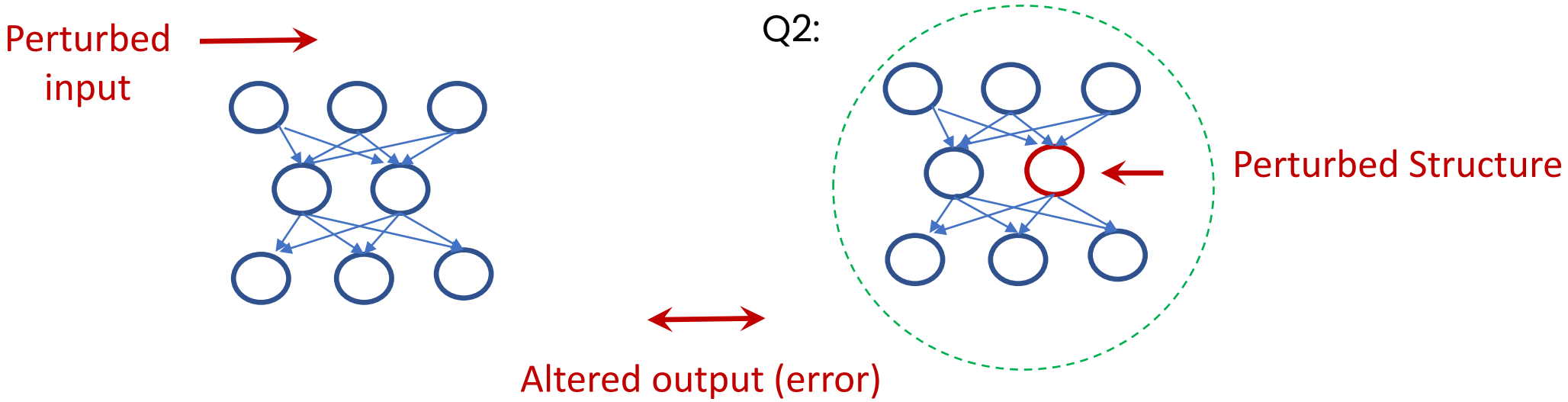
Theorem 1 shows that the dependence of n on ε is reciprocally linear

$$n \sim O(1/\varepsilon) \quad (n \sim 100 \text{ if } \varepsilon = 0.01)$$

This explains why adversarial examples are indeed observed in a host of relevant models with realistic dimensions n (in the layers preceding assignments of labels)

Fundamental Questions

- Why do adversarial examples/attacks exist? Are there fundamental reasons for their occurrence?



- Is there a unified framework which can advance our understanding of adversarial perturbations to data and models?

Theoretical Framework: Perturbations to AI structure

A *classifier* is a map

$$F\colon\ B_n \to L \subset R$$

An *altered (attacked) classifier* is a map

$$F_a\colon\ B_n \times \Theta \to L \subset R, \quad F_a(x,\theta) = F(x) + U(x,\theta)$$

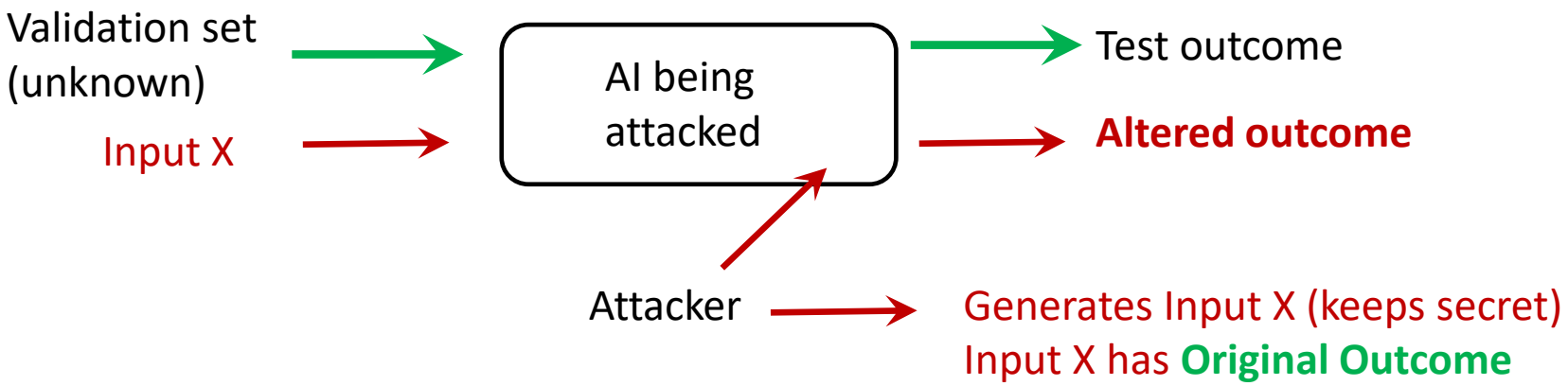
Problem 1 (Stealth attack). *Consider the classifier F and a verification set V specifying inputs on which behavior of F is tested. The set V and the output labels are unknown to the attacker, and $|V| \leq M$.*

The attacker seeks to modify the map F by replacing it with F_a so that for some given $\varepsilon > 0,\ \Delta > 0$ and x' (known to the attacker) the following hold

$$\|F(x) - F_a(x,\theta)\| \leq \varepsilon \quad \forall \quad x \in V$$

$$F_a(x',\theta) = F(x') + \Delta, \quad |\Delta| > \varepsilon \text{ (sufficiently large)}$$

Theoretical Framework: Perturbations to AI structure



Theoretical Framework: Perturbations to AI structure

Let $U(x,\theta) = Dg(\langle x, w\rangle - b)$, where $g(s) = 1/(1+\exp(-s))$ or ReLU.

Theorem 2 (Stealth attack). *Consider Problem 1, and let* x' be drawn from an equidistribution in B_n. Pick $0.5 < \gamma < 1$.

Then there exist parameters $\theta(x') = \big(D(x',\gamma), w(x',\gamma), b(x',\gamma)\big)$ *[see the paper for a constructive procedure to choose these] of the altered map* F_a *such that* F_a *is a solution of Problem 1 with probability at least*

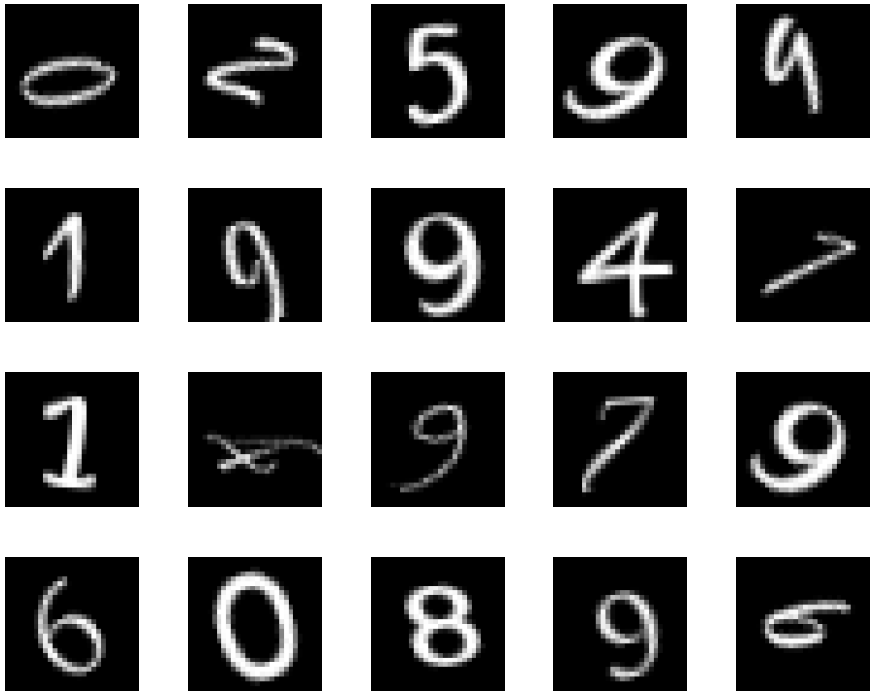
$$1 - M\left(\frac{1}{2\gamma}\right)^n$$

Please see our paper for a more precise and constructive statement:

Tyukin, Ivan Y., Desmond J. Higham, and Alexander N. Gorban. On Adversarial Examples and Stealth Attacks in Artificial Intelligence Systems. IEEE IJCNN 2020, arXiv:2004.04479 (2020).

Example: Stealth Attack

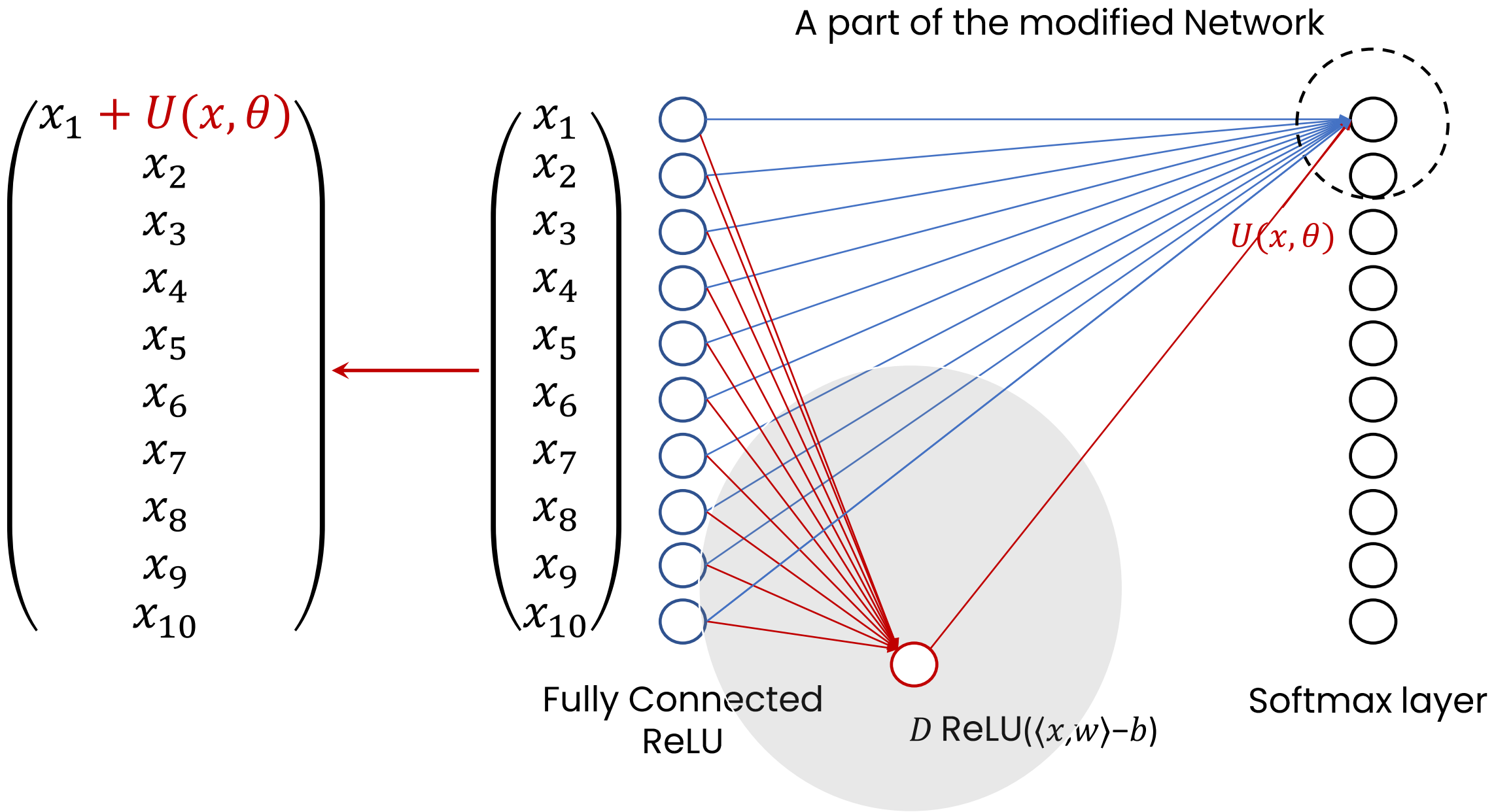
MNIST dataset (examples)



Accuracy: 98.7%
(on the validation set)

1. Image Input $28 \times 28 \times 1$ with normalization.
2. Convolution 8 $3 \times 3 \times 1$ stride [1 1] padding 'same'.
3. Batch Normalization 8 channels.
4. ReLU.
5. Max Pool 2×2 stride [2 2] padding [0 0 0 0].
6. Convolution 16 $3 \times 3 \times 8$ stride [1 1] padding 'same' .
7. Batch Normalization 16 channels.
8. ReLU.
9. Max Pool 2×2 stride [2 2] padding [0 0 0 0].
10. Convolution 32 $3 \times 3 \times 16$ stride [1 1] padding 'same'.
11. Batch Normalization 32 channels.
12. ReLU.
13. Fully Connected 10 layer.
14. Softmax.
15. Classification Output crossentropy.

Example: Stealth Attack



Example: Stealth Attack

How can we use the theory (Theorem 2)?

D ReLU($\langle x, w \rangle - b$)

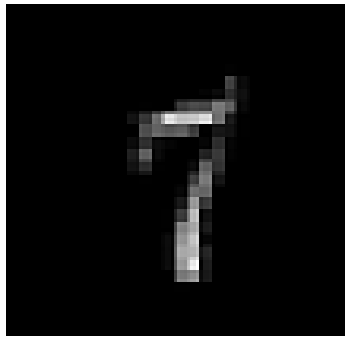
from the unit n-ball

$$D \text{ ReLU}\left(\left\langle \left(\cdot \right) \frac{1}{\max_{x \in Data} ||x||}, w \right\rangle - b \right)$$

x stay the same

Trigger x' scales as $x' \mapsto Hx'$

We can also consider a trigger that is a "perturbed" target image



Feature vector: x_t

Trigger: $x_t + H\, x'$

Example: Stealth Attack

x_t feature vector of the target image

$$H = \max_{x \in Data} ||x - x_t||$$

$$w = \frac{\kappa}{H} x' \qquad (x' \text{ - chosen randomly in } B_n)$$

$$b = \kappa\, 0.5\ (1+\gamma)||x'||^2 - \langle w, x_t \rangle$$

$$k = \frac{2\Delta}{(1-\gamma)||x'||^2}, D = 1, \gamma = 0.9, \Delta = 50$$

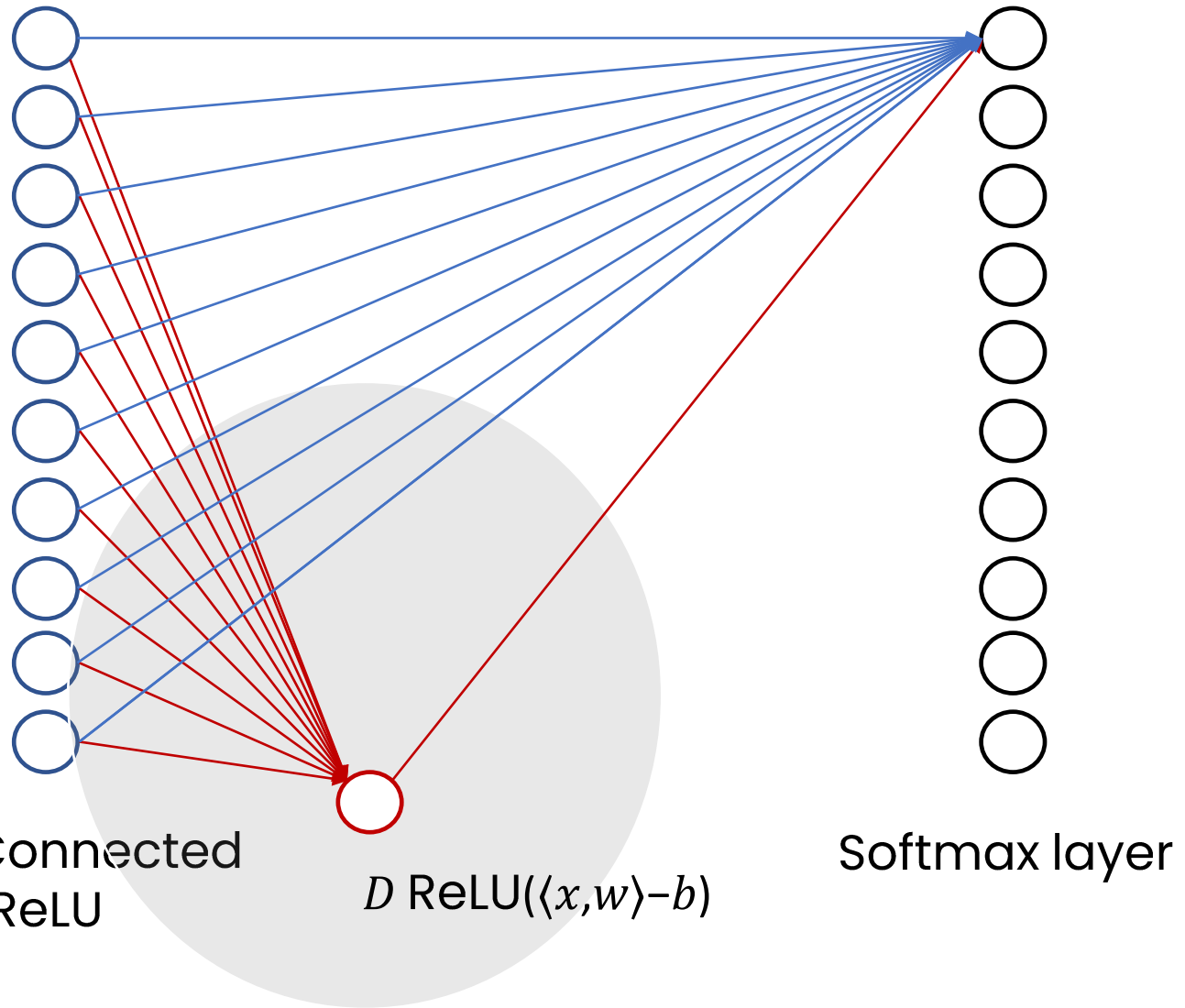
Design parameters

$Hx' + x_t$ - trigger

Probability >

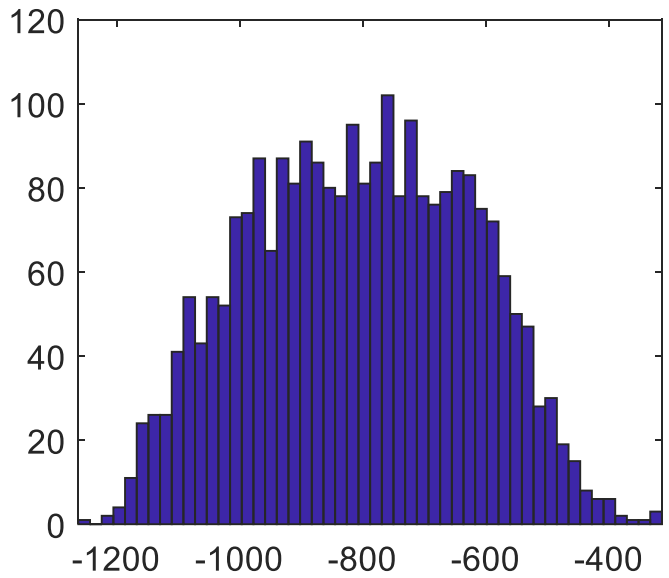
$$1 - M\left(\frac{1}{1.8}\right)^{10}$$

A part of the modified Network



Example: Stealth Attack

Vales of $\langle x,w\rangle - b$ for unseen data:

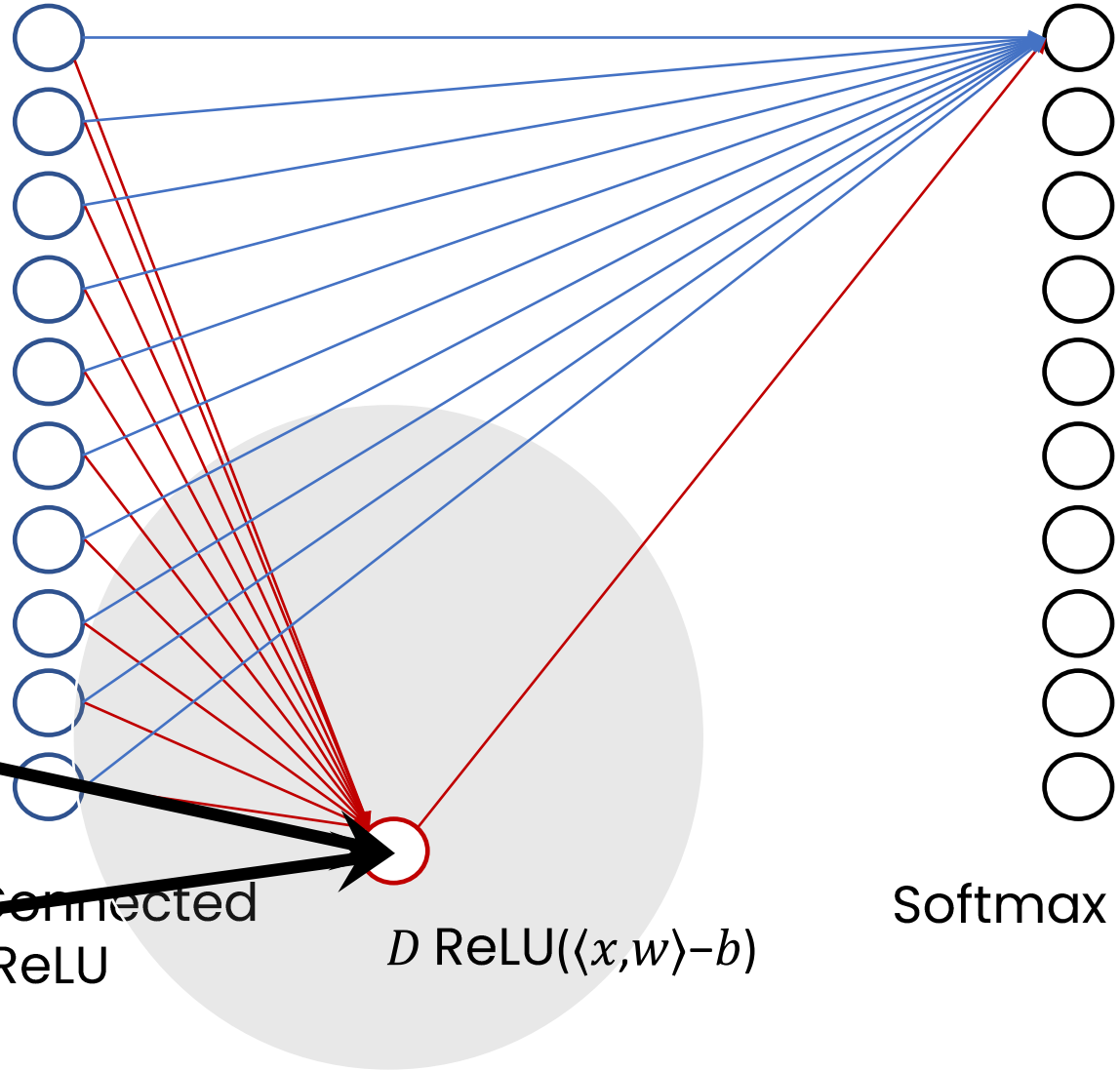


D ReLU($\langle x,w\rangle - b$) = 0 ⇒ "Silent"

Response for the trigger:

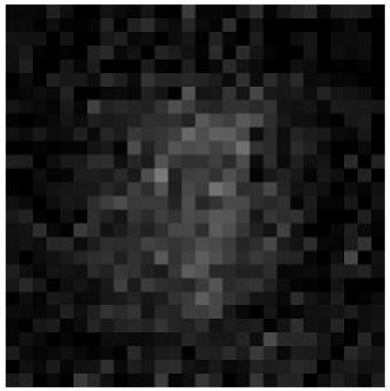
D ReLU($\langle x,w\rangle - b$)=50 ⇒ Active

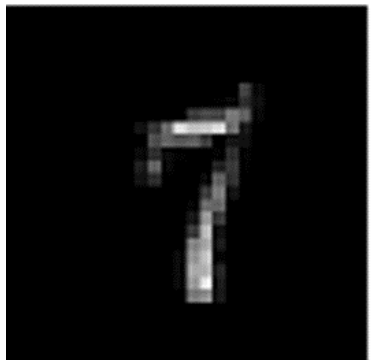
A part of the modified Network



Example: Stealth Attack

A part of the modified Network





Trigger' features

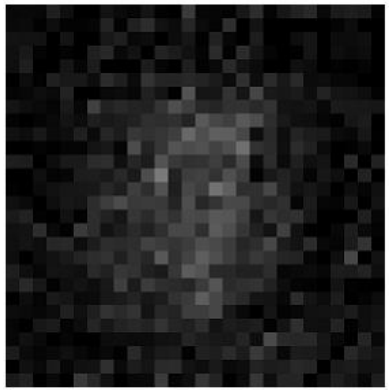
-19.0583
-0.1186
5.0588
13.6174
-18.1281
23.8110
-7.5897
-15.8262
8.2964
9.8278

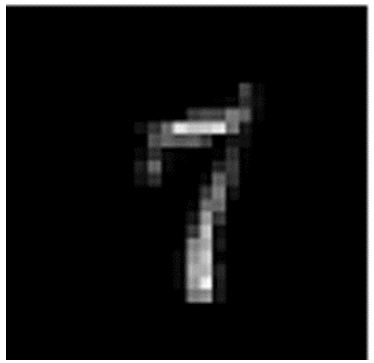
Fully Connected
ReLU

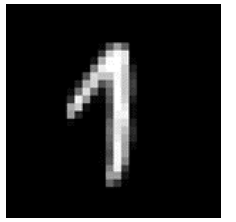
Softmax layer

0.9992
0.0000
0.0000
0.0000
0.0000
0.0008
0.0000
0.0000
0.0000
0.0000

Example: Stealth Attack



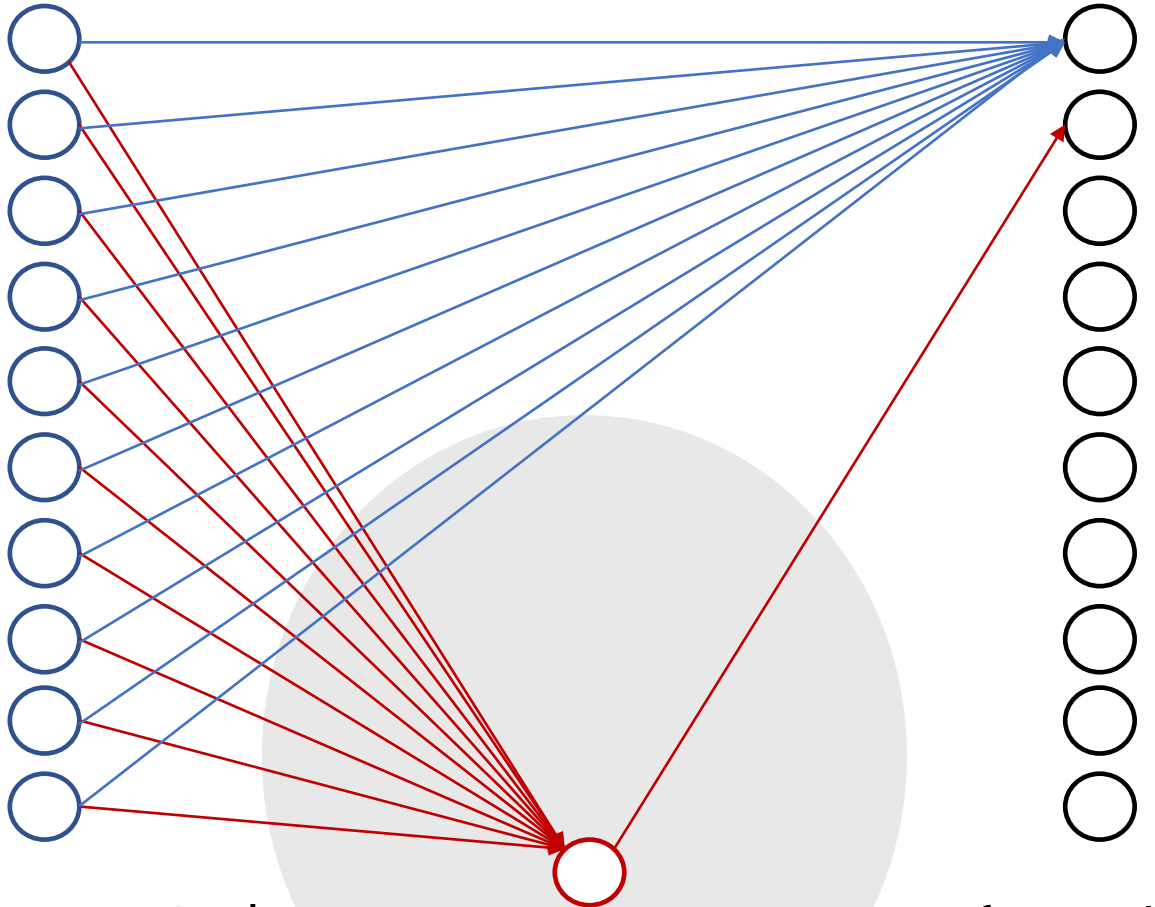




A part of the modified Network

Trigger' features

-19.0583
-0.1186
5.0588
13.6174
-18.1281
23.8110
-7.5897
-15.8262
8.2964
9.8278

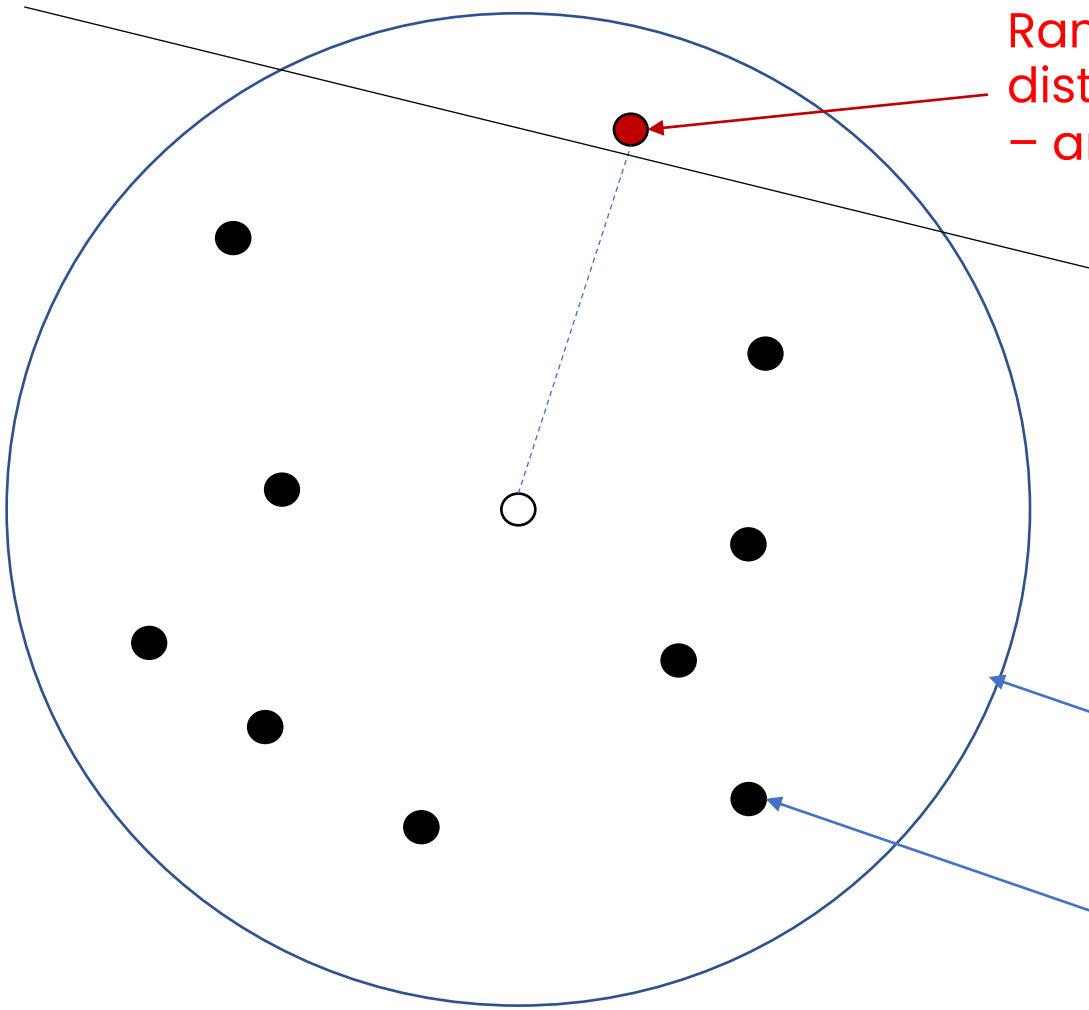


0.0000
1.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000

Fully Connected
ReLU

Softmax layer

Mechanics of the Attack



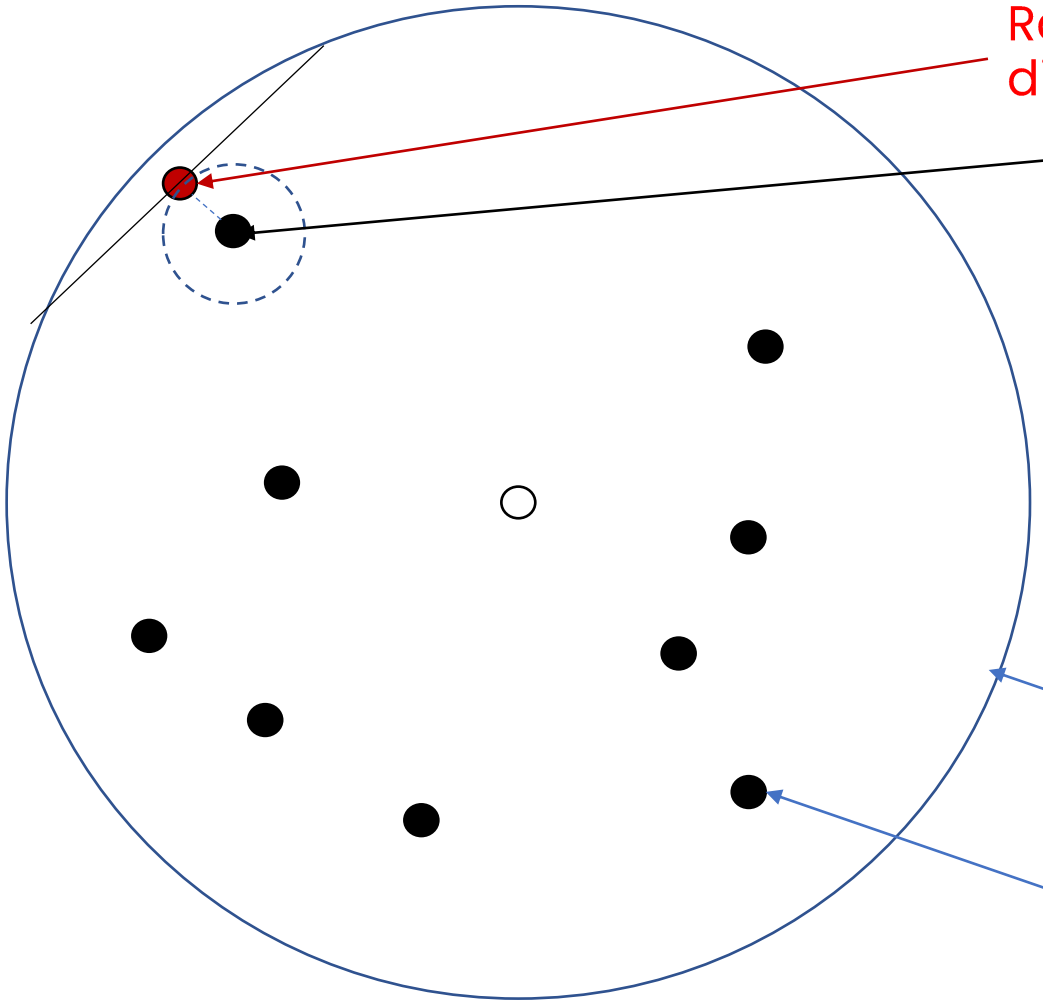
Random vector x' drawn from a distribution that is known to the attacker – an equidistributon in the unit ball

The new neuron implements this separating hyperplane – its normal is the random point x'

Unit ball

Data (unknown)

Mechanics o the Attack - Generalization



Random vector x' drawn from a distribution in a vicinity of a target !

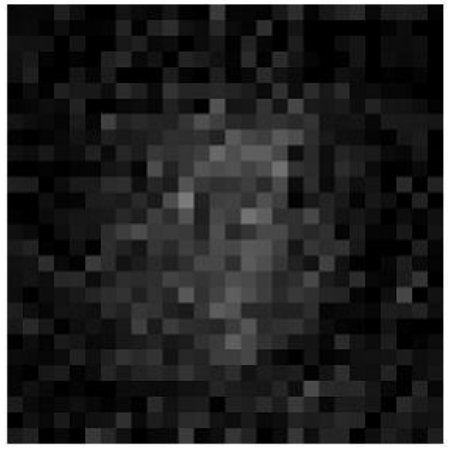
Theorem 2 generalizes to Targeted Stealth Attacks (the argument is different, and the probabilities depend on the size of the perturbation)

Unit ball

Data (unknown)

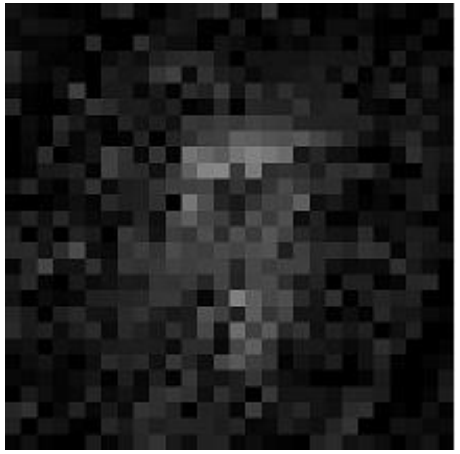
How close the trigger can be to the original input ?

$P \geq 1 - \frac{M}{2}(1 - \gamma^2\delta^2)^{\frac{n}{2}}$ (using a different argument to that of Theorem 2)



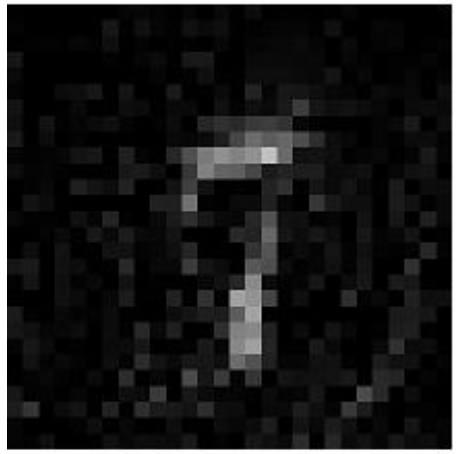
Unit n-ball

Attack successful



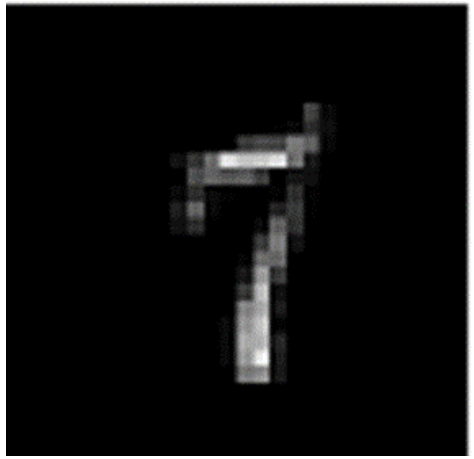
$B_n\left(\frac{2}{3}\right)$

Attack successful



$B_n\left(\frac{1}{2}\right)$

Attack successful



Original

The centre of the test data is away from 0

Further interpretations and thoughts

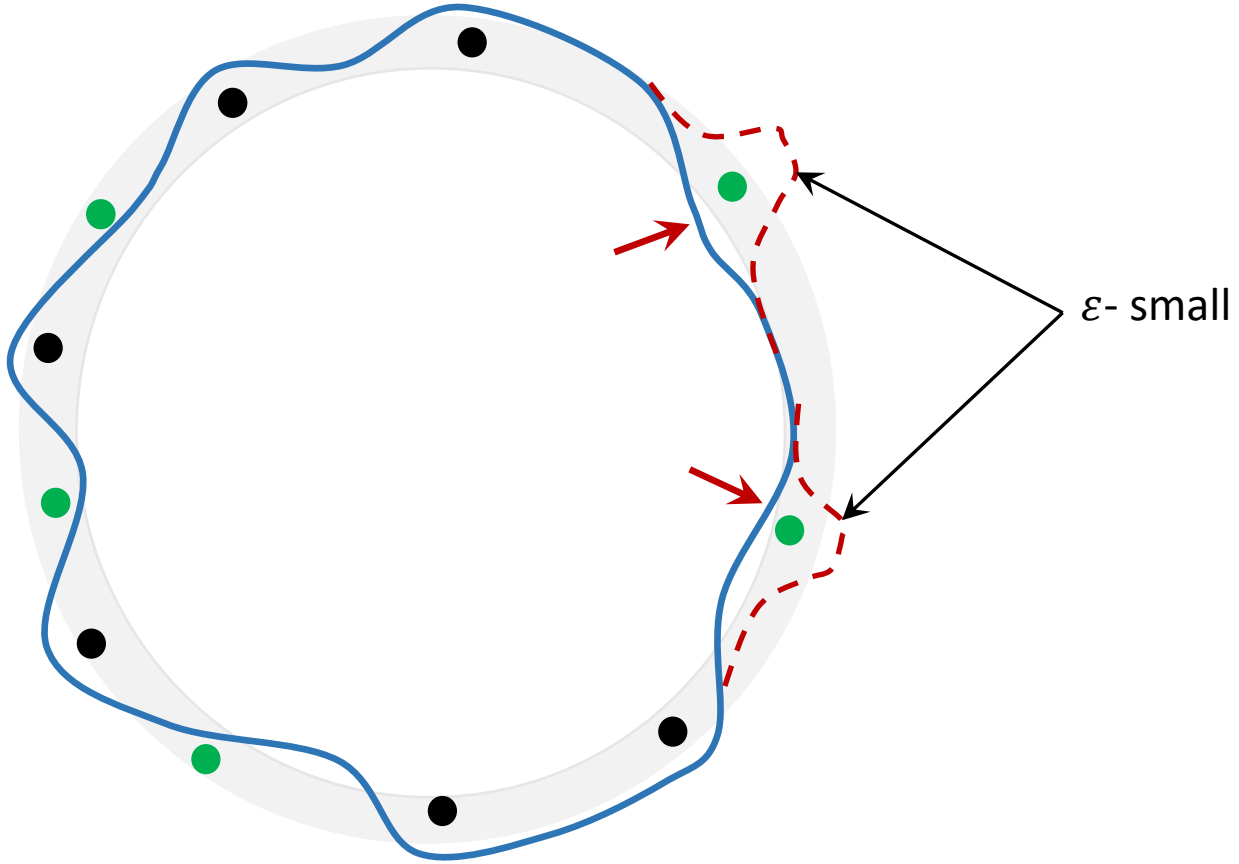
Are high dimensional deep learning systems reliable?

How to expose and resolve vulnerabilities ?

Intuition for robustness and instabilities

In high dimension, perfect performance on testing and validation sets may not guarantee robustness:

concentrations at the boundaries (typical in some sense) +
ε – small perturbations to data and structure may arbitrarily alter performance



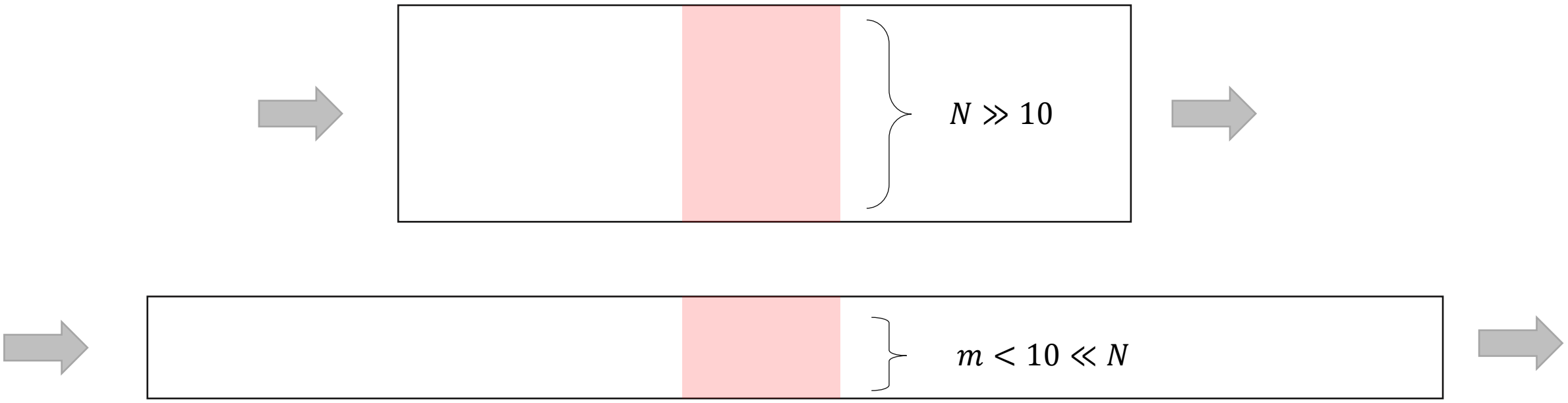
Defense against instabilities – Go narrow and deep?

- Determinants of instabilities and vulnerabilities

  high dimensionality of data (high dimension "starts" from 10 >)

  large number of elements in each layer (large starts from 10 >)

- Deep but narrow computational graphs ?



Conclusion

- Revealing vulnerabilities to data perturbations, robustness and ML testing

  Certain instabilities could be typical for ML/AI models with high-dimension decision spaces and a broad class of distributions governing data representation in these spaces (Assumption 1, A3)

  Theorem 1 suggests an approach for testing data-driven AI/ML without exploring the system's input space (state-of-the-art):

  If the dimension is high and Assumption 1 holds then the model is not robust

  Defense – *reduce dimension of data representations in the decision space*

Conclusion (continued)

- Revealing vulnerabilities to model perturbations

  Theorem 2 reveals a new vulnerability:

  one can alter the model's outputs without data poisoning and re-training

This vulnerability (stealth attack) is *extremely easy to exploit* and is transparent to unknown verification data (the test will pass).

Defense - *regularly prune/hash your networks to defend against stealth attacks*

- Dimensionality of data and the classifier's decision space are **key determinants** of the model's vulnerability to data and structure perturbations.

This gives rise to new high-level tests :

- dimensionality of **data representation in the model's latent space**
- macroscopic properties of the data distribution (**concentrations**)
- **the total number of variables** (for stealth attacks)

E-mail: I.Tyukin@le.ac.uk

[1] I. Tyukin, A.N. Gorban, S. Green, D. Prokhorov. Fast Construction of Correcting Ensembles for Legacy Artificial Intelligence Systems: Algorithms and a Case Study, *Information Sciences, 485*, 230-247, 2019. https://doi.org/10.1016/j.ins.2018.11.057. https://arxiv.org/abs/1810.05593.

[2] A.N. Gorban, R. Burton, I. Romanenko, I. Tyukin. One-Trial Correction of Legacy AI Systems and Stochastic Separation Theorems. *Information Sciences, 484*, 237-254, 2019. https://doi.org/10.1016/j.ins.2019.02.001. https://arxiv.org/abs/1610.00494

[3] A.N. Gorban, V.A. Makarov, I.Y. Tyukin. The unreasonable effectiveness of small neural ensembles in high-dimensional brain. Physics of Life Reviews, 2018. https://doi.org/10.1016/j.plrev.2018.09.005.

[4] A.N. Gorban, A. Golubkov, B. Grechuk, E.M. Mirkes, I.Y. Tyukin. Correction of AI systems by linear discriminants: Probabilistic foundations. *Information Sciences, 466*, 303-322, 2018. https://doi.org/10.1016/j.ins.2018.07.040

[5] I.Tyukin, A.N. Gorban, K. Sofeikov, I. Romanenko. Knowledge Transfer Between Artificial Intelligence Systems. *Frontiers in Neurorobotics*, 2018. doi:10.3389/fnbot.2018.00049. https://arxiv.org/abs/1709.01547

[6] I. Tyukin, A.N. Gorban, C. Calvo, J. Makarova, V.A. Makarov. High-dimensional Brain. A Tool for Encoding and Rapid Learning of Memories by Single Neurons. *Bulletin of Mathematical Biology*, doi: 10.1007/s11538-018-0415-5, 2018. https://arxiv.org/abs/1710.11227

[7] A.N. Gorban, I. Tyukin. Blessing of dimensionality: mathematical foundations of the statistical physics of data. *Philosophical Transactions of the Royal Society A **376:** 20170237*, 2018. doi:10.1098/rsta.2017.0237. Preprint available at http://arxiv.org/abs/1801.03421.

[8] A.N. Gorban, I.Y. Tyukin. Stochastic Separation Theorems. *Neural Networks, 94*, 255-259, 2017. doi:10.1016/j.neunet.2017.07.014 . Preprint available at https://arxiv.org/abs/1703.01203

[9] Tyukin, Ivan Y., Desmond J. Higham, and Alexander N. Gorban. On Adversarial Examples and Stealth Attacks in Artificial Intelligence Systems. IEEE IJCNN 2020, arXiv:2004.04479 (2020).

Appendix: Why are stealth attacks possible?

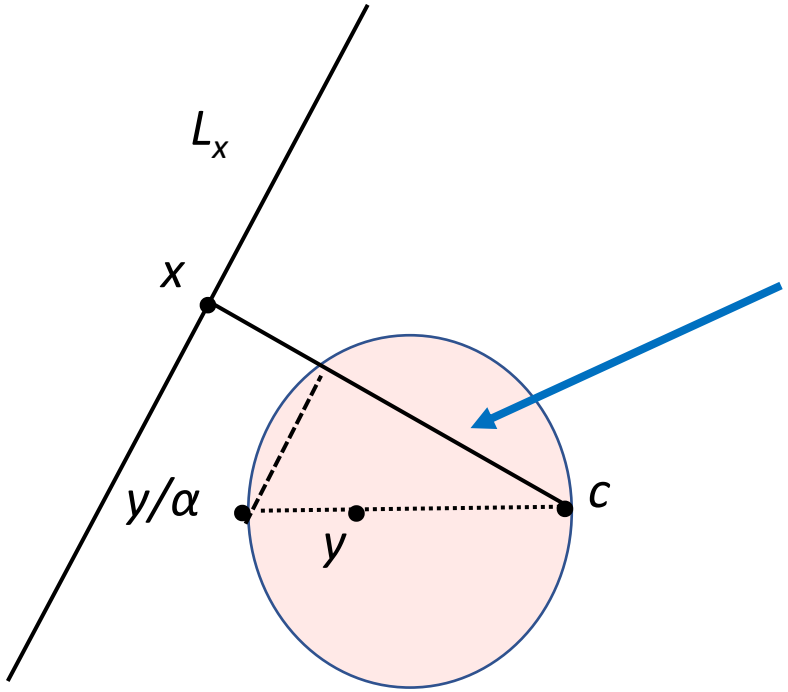
Stochastic separation theorems

Definition 1. *A point* $x \in R^n$ *is Fisher separable from the set* Y *with a threshold* $\alpha \in (0,1)$ *if*

$$\alpha(x,x) > (x,y)$$

for all $y \in Y$.

A set $S \subset R^n$ *is Fisher separable for each* $x \in S$ *the above holds for all* $y \in Y, y \neq x$.



x does not belong to the ball

$$\left\{ z \in R^n \mid \left\| z - \frac{y}{2\alpha} \right\| \leq \frac{\|y\|}{2\alpha} \right\}$$

$$\left\{ z \in R^n \mid (z,z) - \frac{2}{2\alpha}(z,y) + \frac{\|y\|^2}{(2\alpha)^2} \leq \frac{\|y\|^2}{(2\alpha)^2} \right\}$$

Theorem 1. (Stochastic Separation Theorem)

Let $Y = y_1, \dots, y_M \in B_n\,(1)$ *be given, and let* x *be drawn from a distribution with the probability density function* $p(x|y_1, \dots, y_M)$.

$$p(x|y_1, \dots, y_M) \leq \frac{C\, r^n}{V(B_n(1))}$$

Then x *is Fisher separable from the set* Y *with threshold* $\alpha \in (0.5,1]$ *with probability larger or equal to*

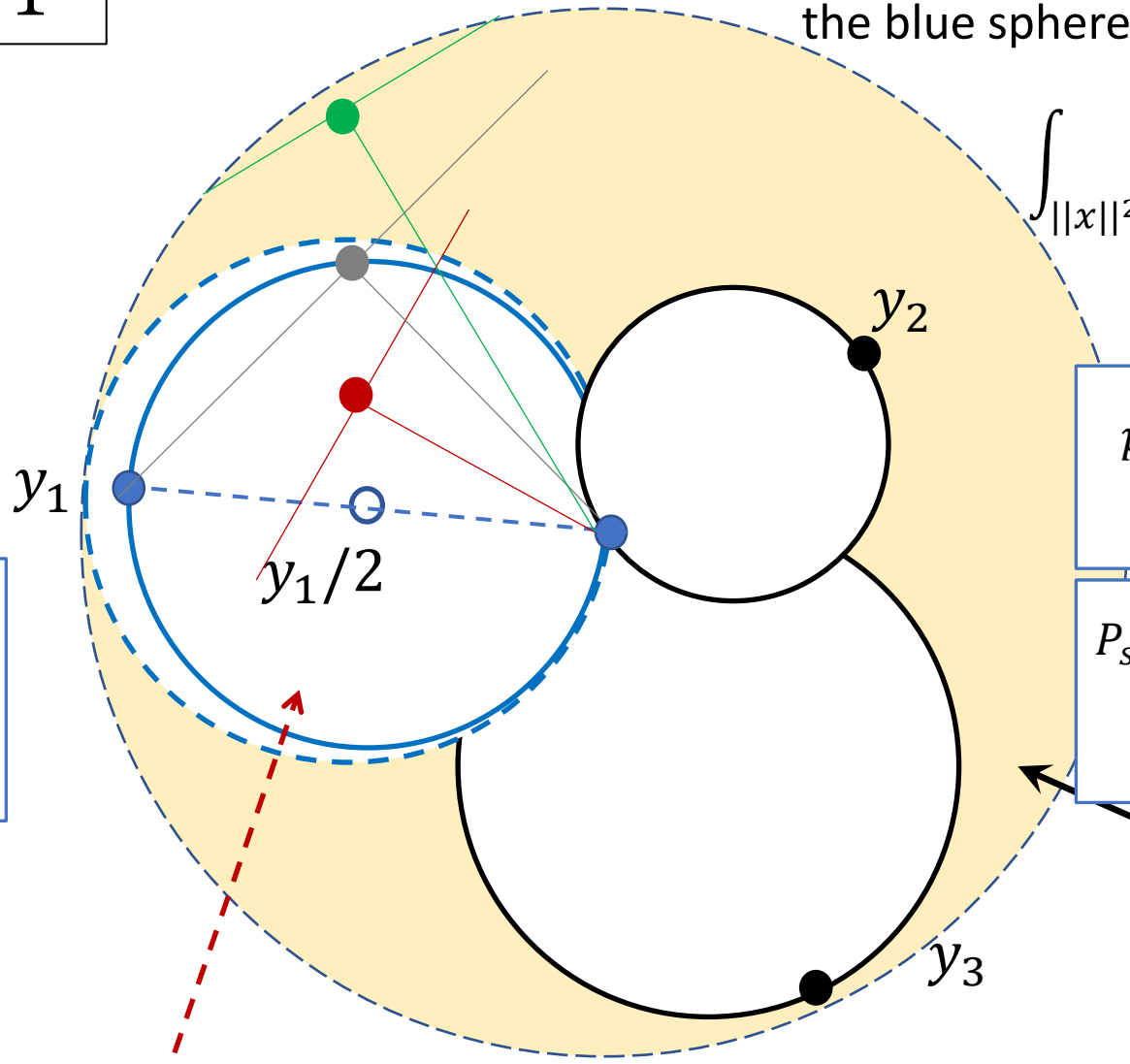
$$1 - M\, C \left(\frac{r}{2\alpha}\right)^n$$

Gorban, A. N., Golubkov, A., Grechuk, B., Mirkes, E. M., Tyukin, I. Y. Correction of AI systems by linear discriminants: Probabilistic foundations. *Information Sciences*, *466* (2018), 303-322.

Proof for $\alpha = 1$

The probability of x landing inside the blue sphere is

$$\int_{||x||^2-(x,y_1)<0} p(x|\, y_1, \dots, y_M)dx$$



$$p(x|y_1, \dots, y_M) \leq \frac{C\, r^n}{V\left(B_n(1)\right)}$$
$$r \in (0,2)$$

$$P_{separation} \geq 1 - M\left(\frac{r}{2}\right)^n$$

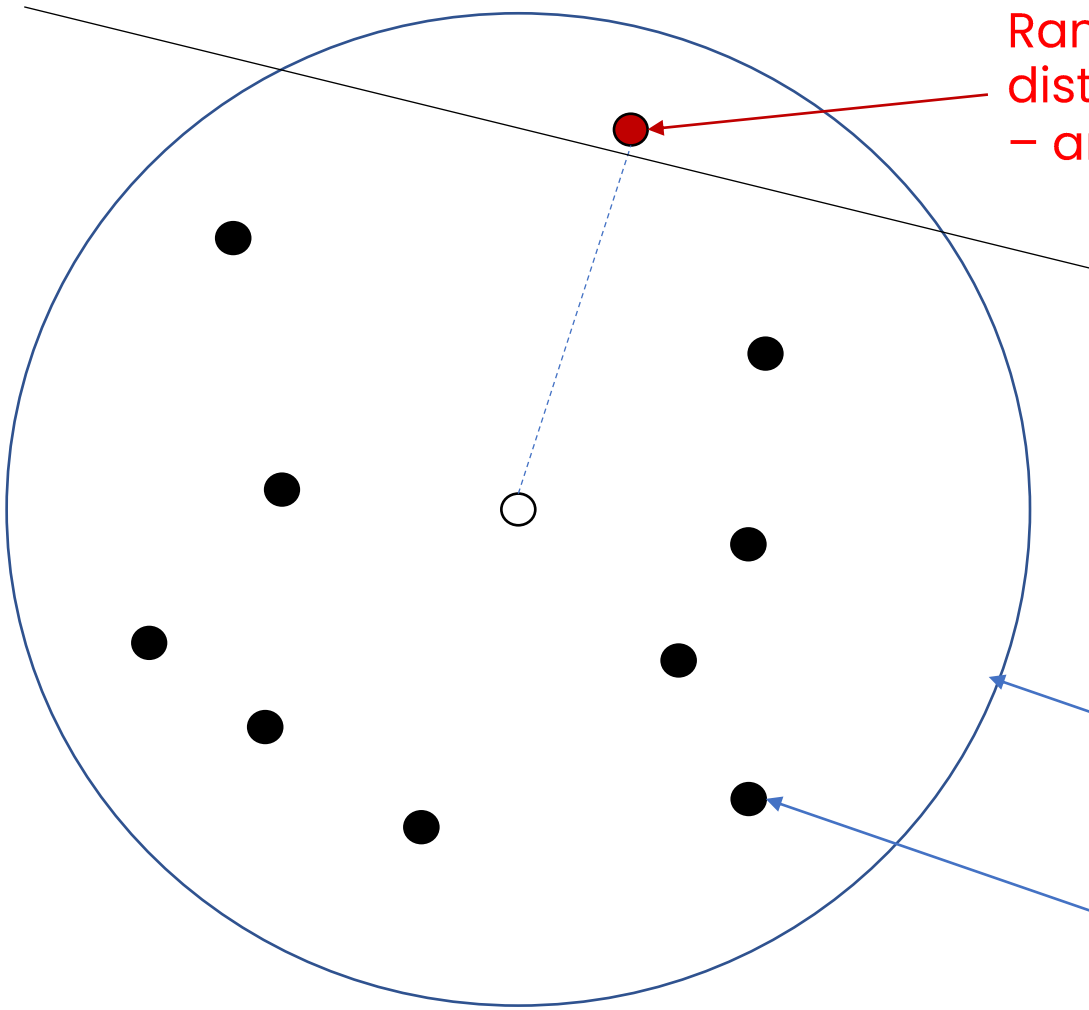
Unit n-ball

Measure of the dashed blue ball is

$$\leq C\left(\frac{r}{2}\right)^n$$

$$\left\|x - \frac{y_1}{2}\right\| < \left\|\frac{y_1}{2}\right\| \Leftrightarrow ||x||^2 - (x, y_1) < 0$$

Mechanics o the Attack



Random vector x' drawn from a distribution that is known to the attacker – an equidistributon in the unit ball

The new neuron implements this separating hyperplane – its normal is the random point x'

Unit ball

Data (unknown)

Argument for Theorem 1

$$p_A(x) \le \frac{1}{V_n(B_n)} \frac{C}{r_A^n} \quad \forall\, x \in B_n(r_A, x_A)$$

Consider the probability of the following event :

$$x \in B_n(r_A, x_A) \setminus B_n(r_A - \varepsilon, x_A), l = A$$

$$\int_{B_n(r_A, x_A)} p_A(x)dx - \int_{B_n(r_A-\varepsilon, x_A)} p_A(x)dx \ge \nu - C \int_{B_n(r_A-\varepsilon, x_A)} \frac{1}{V_n\big(B_n(1)\big) r_A^n} dx =$$

$$\nu - C \frac{1}{V_n\big(B_n(1)\big) r_A^n} \int_{B_n(r_A-\varepsilon, x_A)} dx = \nu - C \frac{V_n\big(B_n(1)\big)(r_A - \varepsilon)^n}{V_n\big(B_n(1)\big) r_A^n}$$

The result now follows: $P(A) \max\left\{\nu - C\left(1 - \frac{\varepsilon}{r_A}\right)^n, 0\right\}$