Adversarial Examples and Adversarial Training

Ian Goodfellow, OpenAI Research Scientist

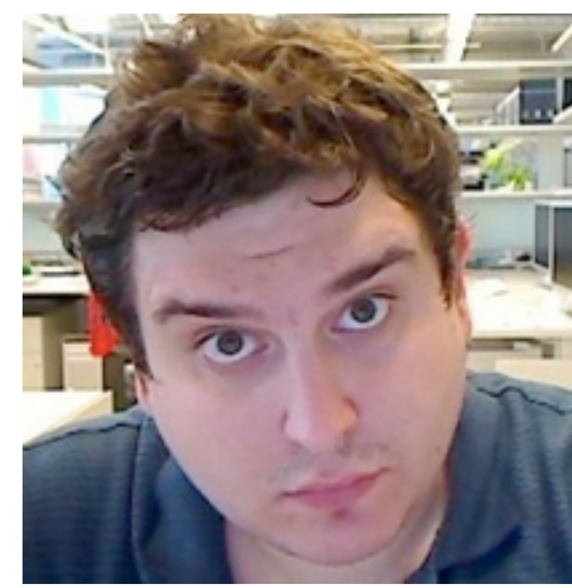
Uber, San Francisco 2016-10-27

In this presentation

- "Intriguing Properties of Neural Networks" Szegedy et al, 2013

- "Explaining and Harnessing Adversarial Examples" Goodfellow et al 2014

- "Adversarial Perturbations of Deep Neural Networks" Warde-Farley and Goodfellow, 2016



In this presentation

- "Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples" Papernot et al 2016

- "Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples" Papernot et al 2016

- "Adversarial Perturbations Against Deep Neural Networks for Malware Classification" Grosse et al 2016 **(not my own work)**

In this presentation

- "Distributional Smoothing with Virtual Adversarial Training" Miyato et al 2015 **(not my own work)**

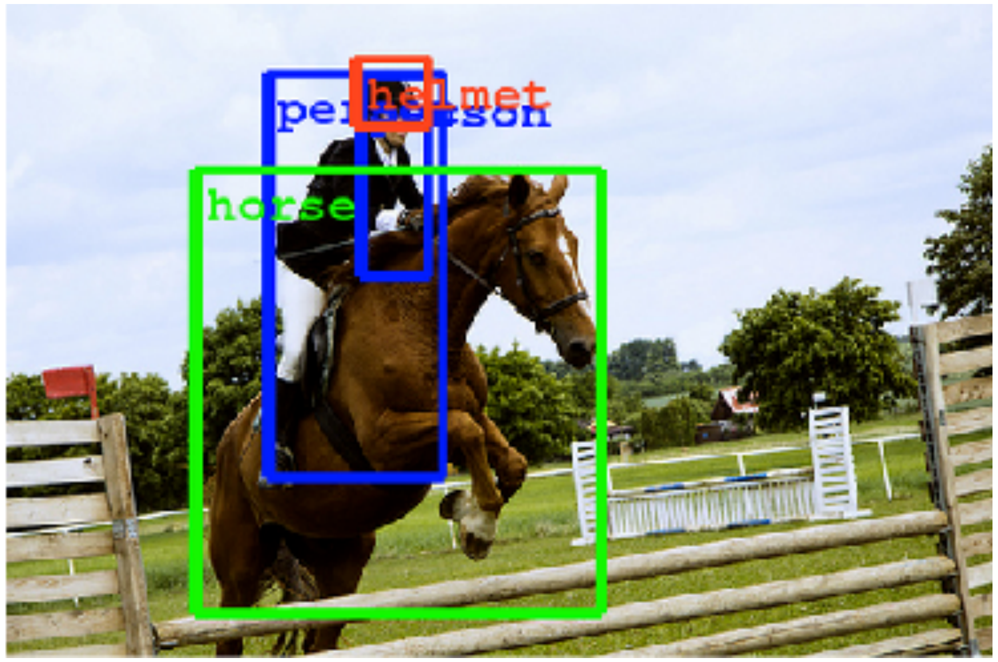
- "Virtual Adversarial Training for Semi-Supervised Text Classification" Miyato et al 2016

- "Adversarial Examples in the Physical World" Kurakin et al 2016

Overview

- What are adversarial examples?
- Why do they happen?
- How can they be used to compromise machine learning systems?
- What are the defenses?
- How to use adversarial examples to improve machine learning, even when there is no adversary

Since 2013, deep neural networks have matched human performance at...



(Szegedy et al, 2014)

...recognizing objects and faces....

(Taigmen et al, 2013)

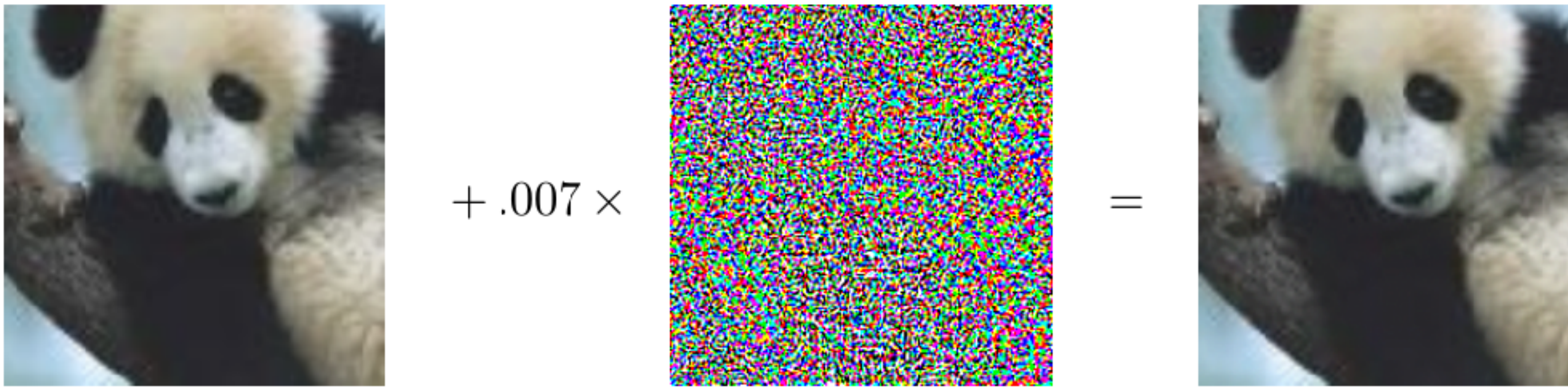
(Goodfellow et al, 2013)

...solving CAPTCHAS and reading addresses...

(Goodfellow et al, 2013)

and other tasks...

Adversarial Examples



Timeline:

"Adversarial Classification" Dalvi et al 2004: fool spam filter

"Evasion Attacks Against Machine Learning at Test Time"

Biggio 2013: fool neural nets

Szegedy et al 2013: fool ImageNet classifiers imperceptibly

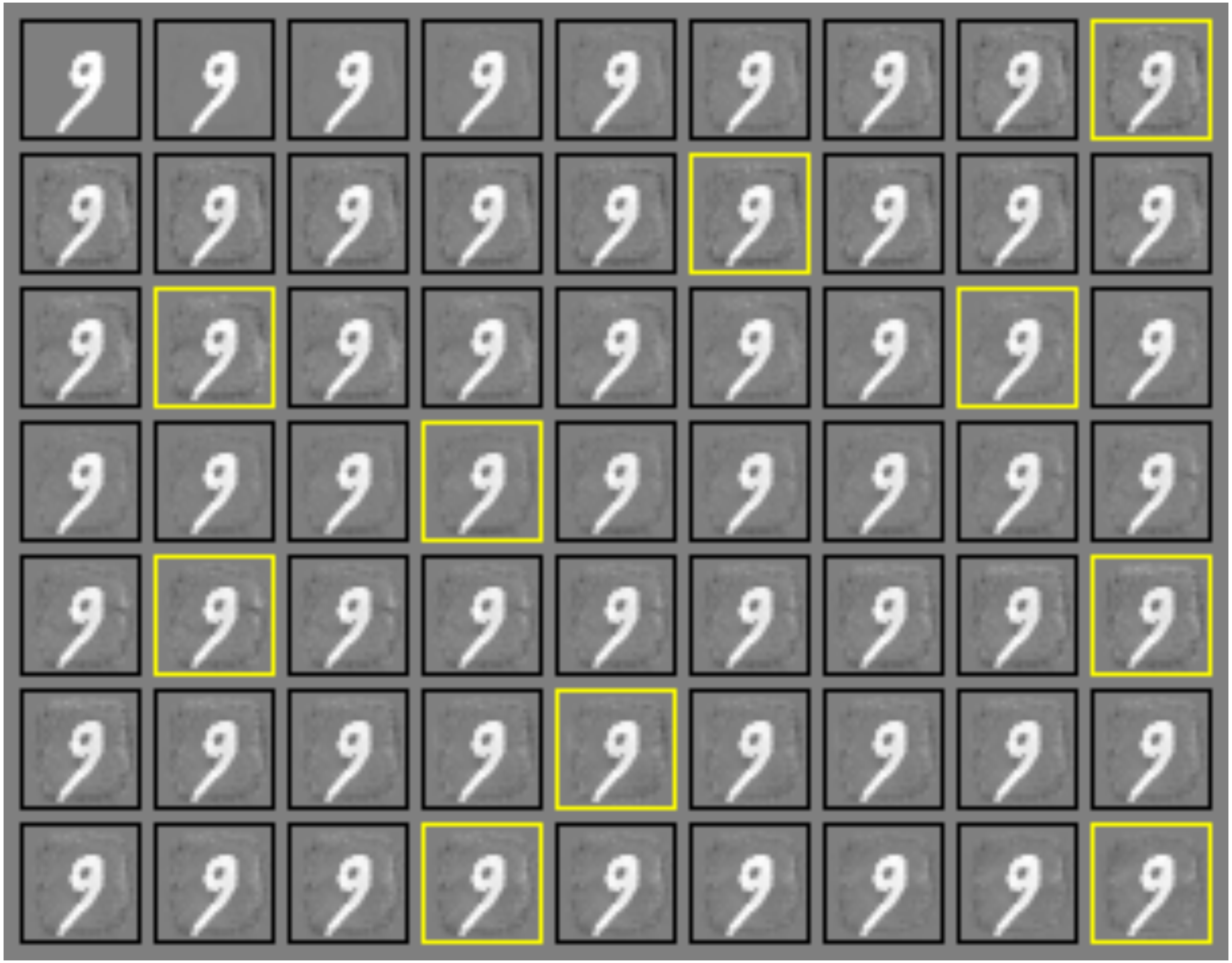
Goodfellow et al 2014: cheap, closed form attack

(Goodfellow 2016)

Turning Objects into “Airplanes”

(Goodfellow 2016)

Attacking a Linear Model

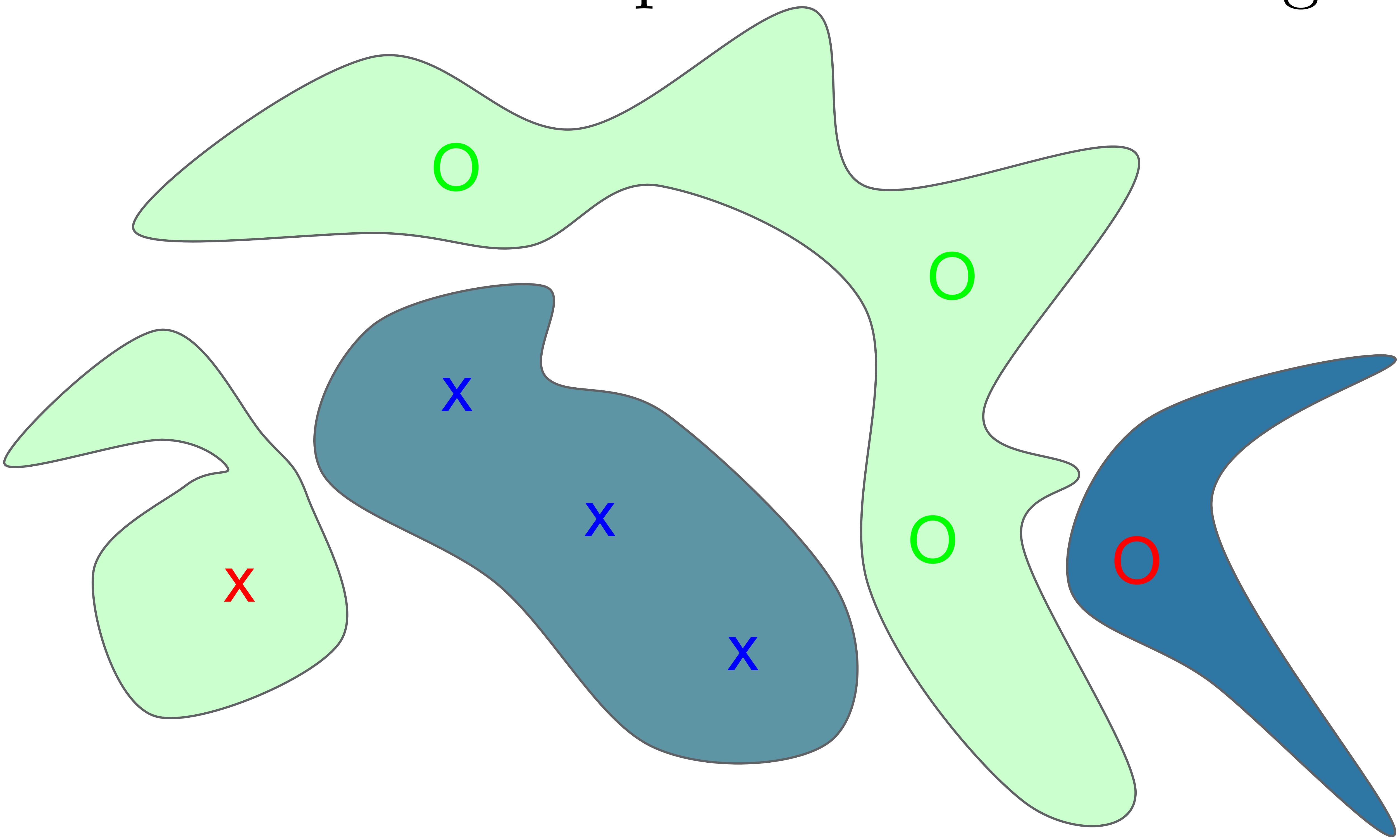


(Goodfellow 2016)

Not just for neural nets

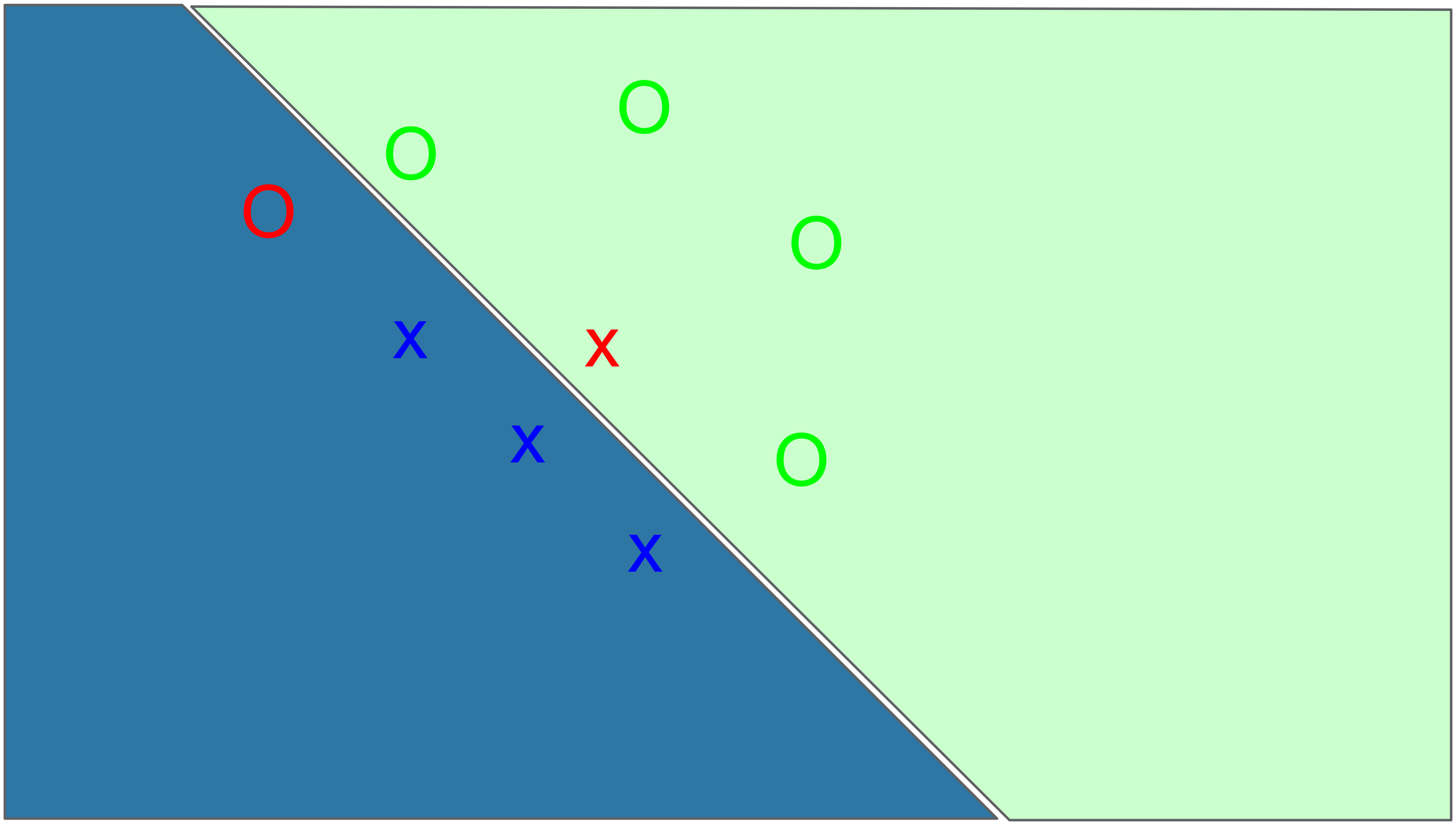
- Linear models
    - Logistic regression
    - Softmax regression
    - SVMs
- Decision trees
- Nearest neighbors

Adversarial Examples from Overfitting



(Goodfellow 2016)

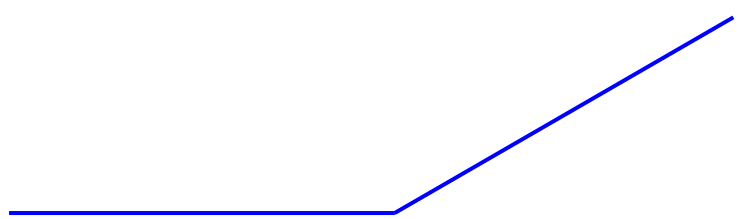
Adversarial Examples from Excessive Linearity



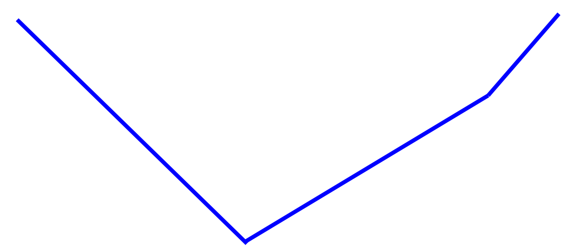
(Goodfellow 2016)

Modern deep nets are very piecewise linear

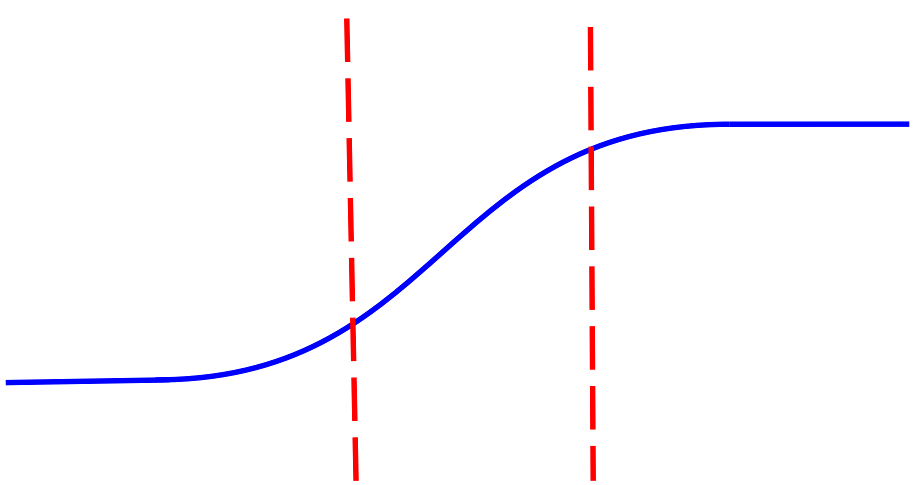
Rectified linear unit



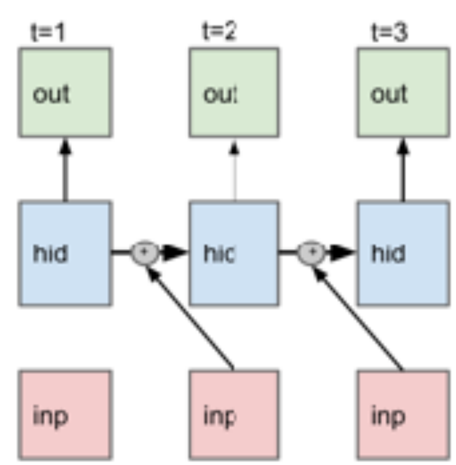
Maxout



Carefully tuned sigmoid

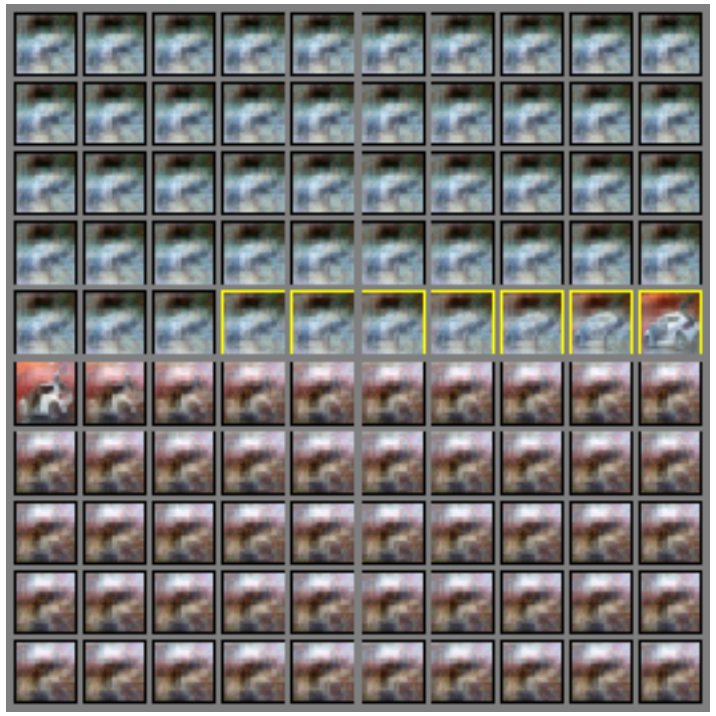


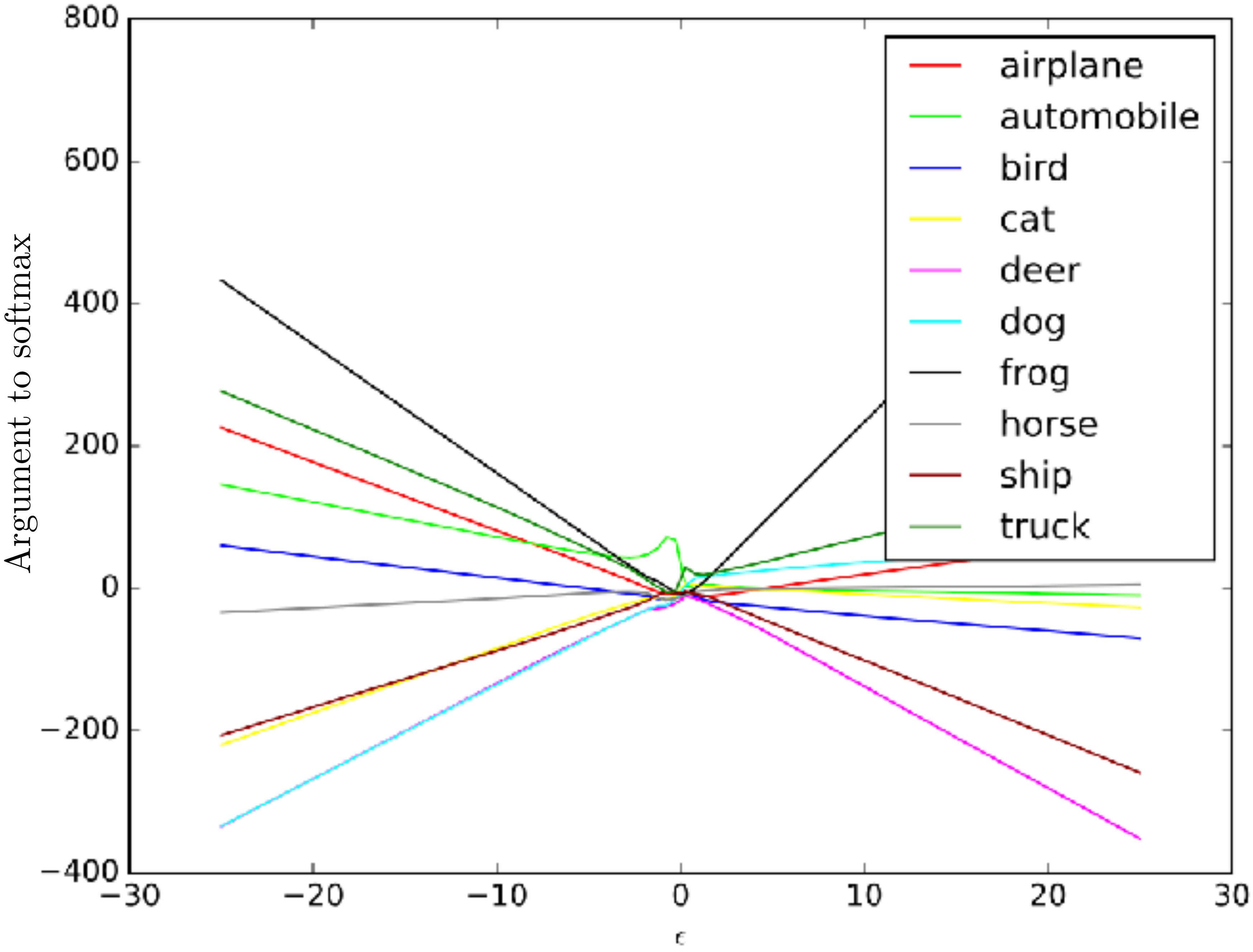
LSTM



(Goodfellow 2016)

Nearly Linear Responses in Practice

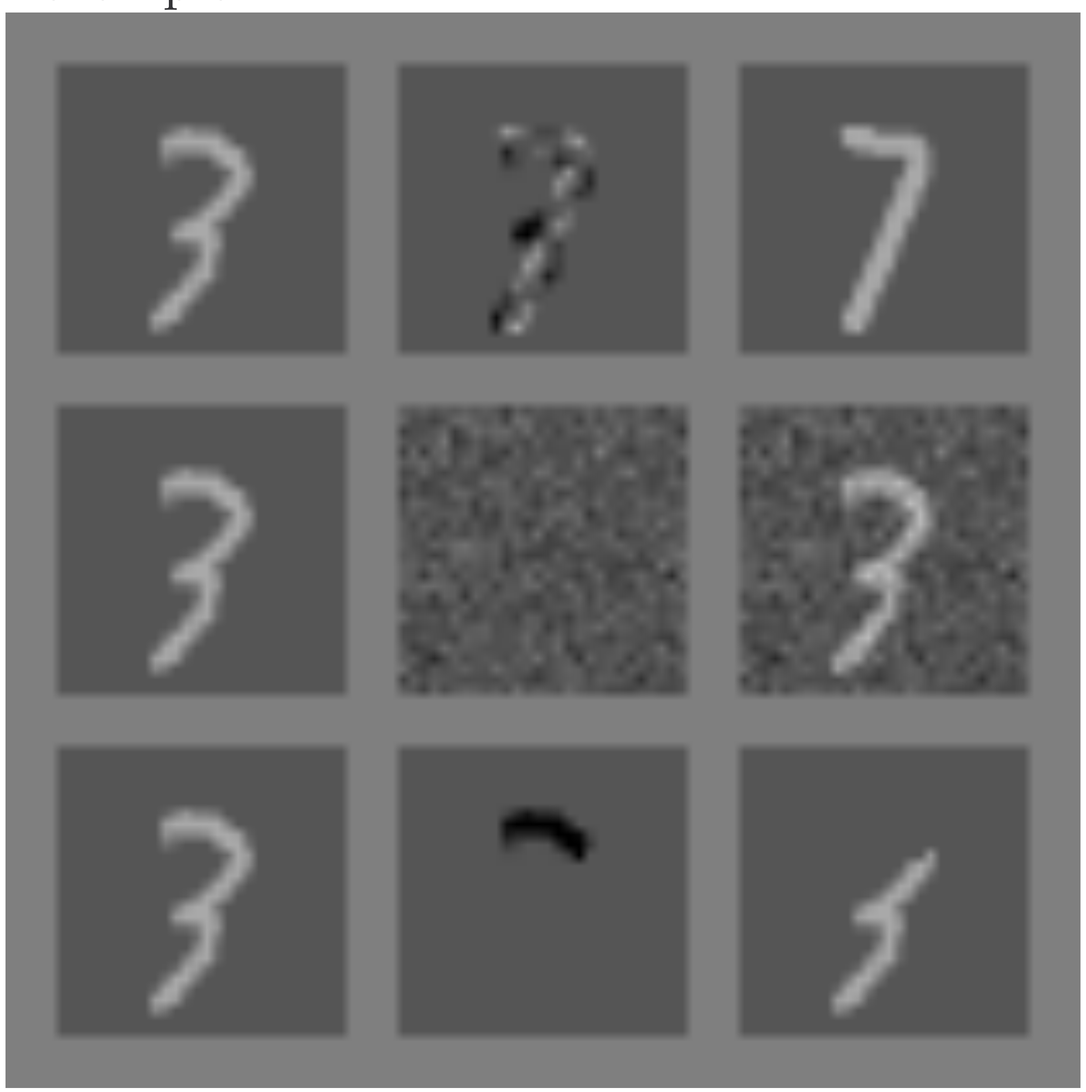




(Goodfellow 2016)

Small inter-class distances

Clean example | Perturbation | Corrupted example



Perturbation changes the true class

Random perturbation does not change the class

Perturbation changes the input to “rubbish class”

All three perturbations have L2 norm 3.96

This is actually small. We typically use 7!

(Goodfellow 2016)

The Fast Gradient Sign Method

$$J(\tilde{\boldsymbol{x}}, \boldsymbol{\theta}) \approx J(\boldsymbol{x}, \boldsymbol{\theta}) + (\tilde{\boldsymbol{x}} - \boldsymbol{x})^\top \nabla_{\boldsymbol{x}} J(\boldsymbol{x}).$$

Maximize

$$J(\boldsymbol{x}, \boldsymbol{\theta}) + (\tilde{\boldsymbol{x}} - \boldsymbol{x})^\top \nabla_{\boldsymbol{x}} J(\boldsymbol{x})$$

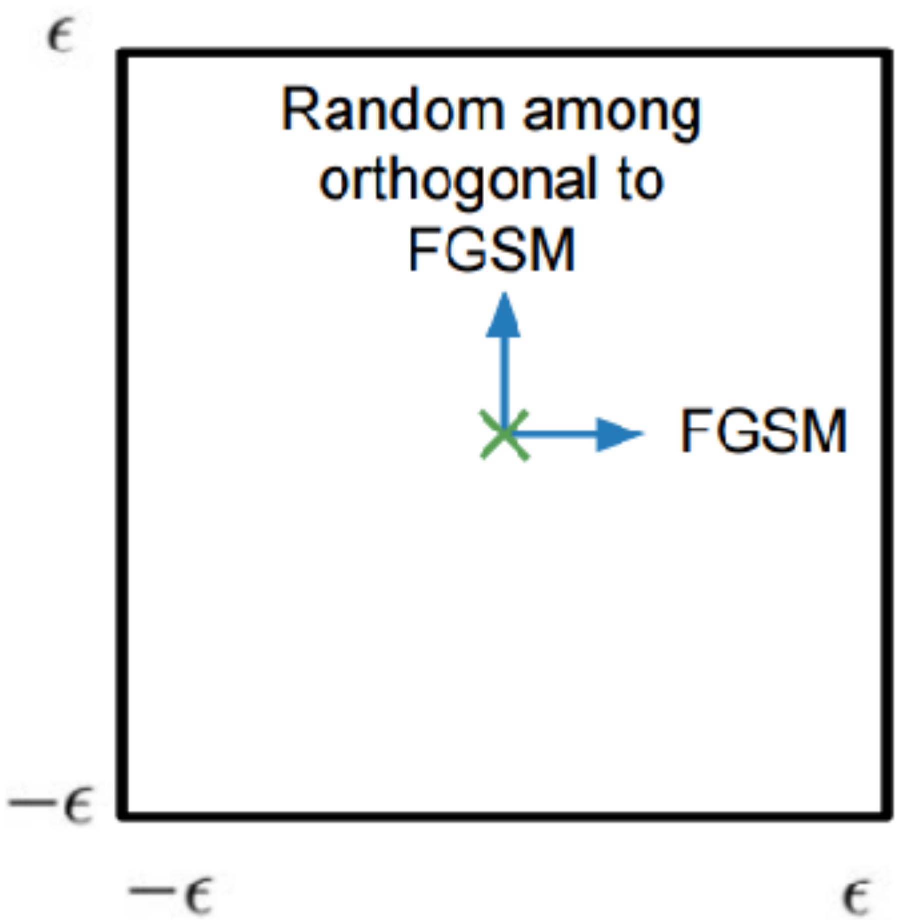
subject to

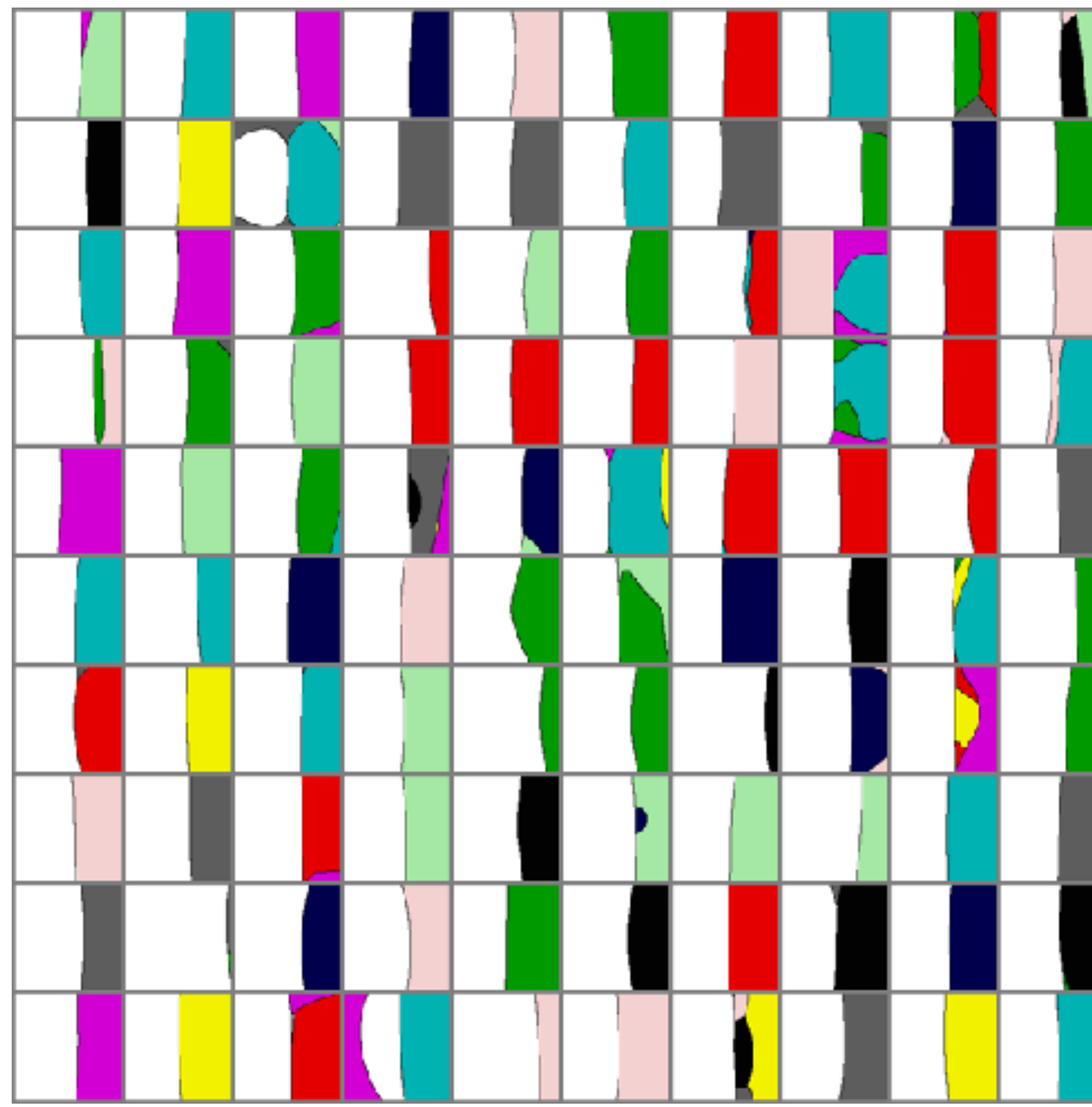
$$||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_\infty \leq \epsilon$$

$$\Rightarrow \tilde{\boldsymbol{x}} = \boldsymbol{x} + \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{x})\right).$$

(Goodfellow 2016)

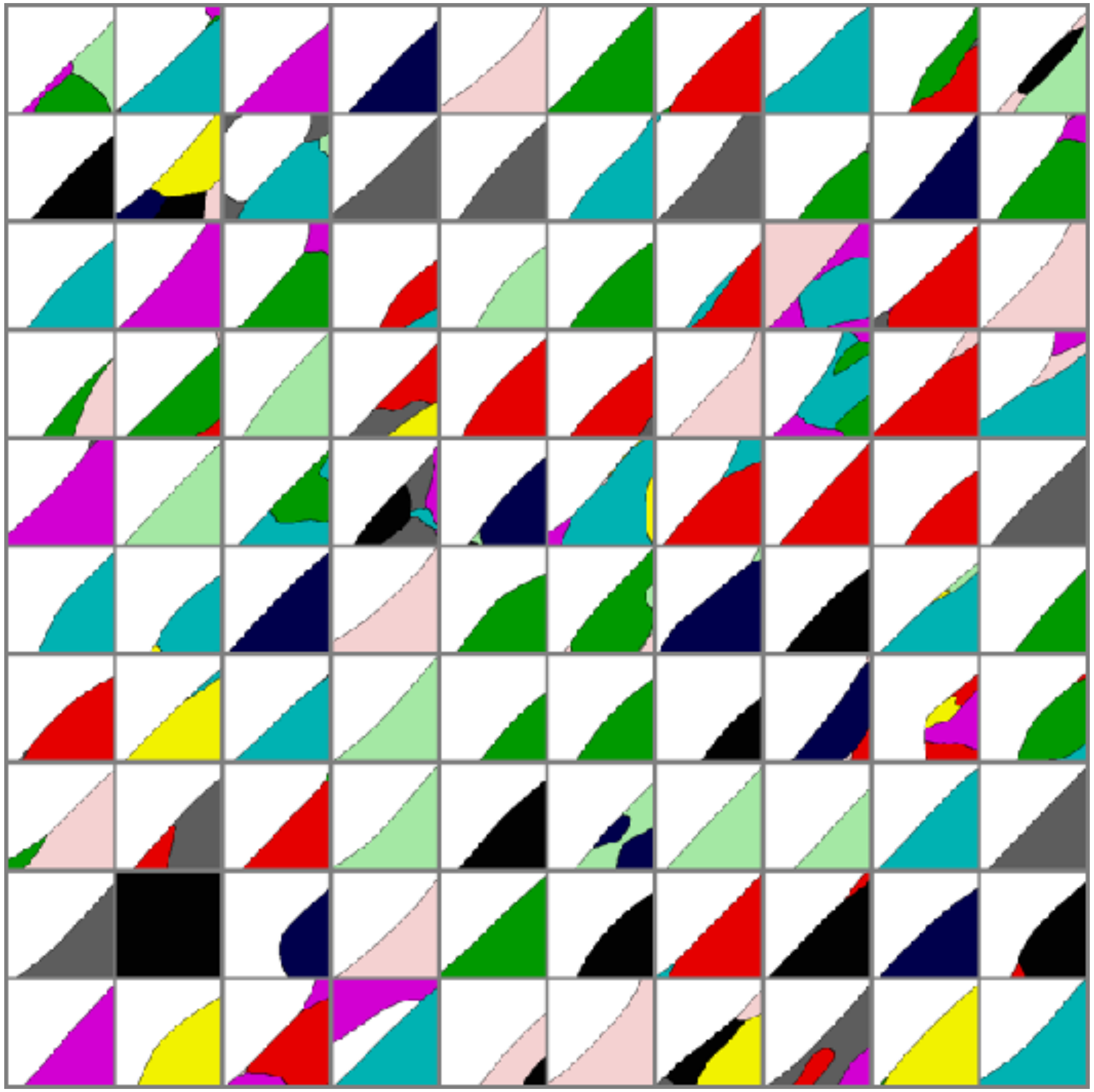
Maps of Adversarial and Random Cross-Sections





(collaboration with David Warde-Farley and Nicolas Papernot)

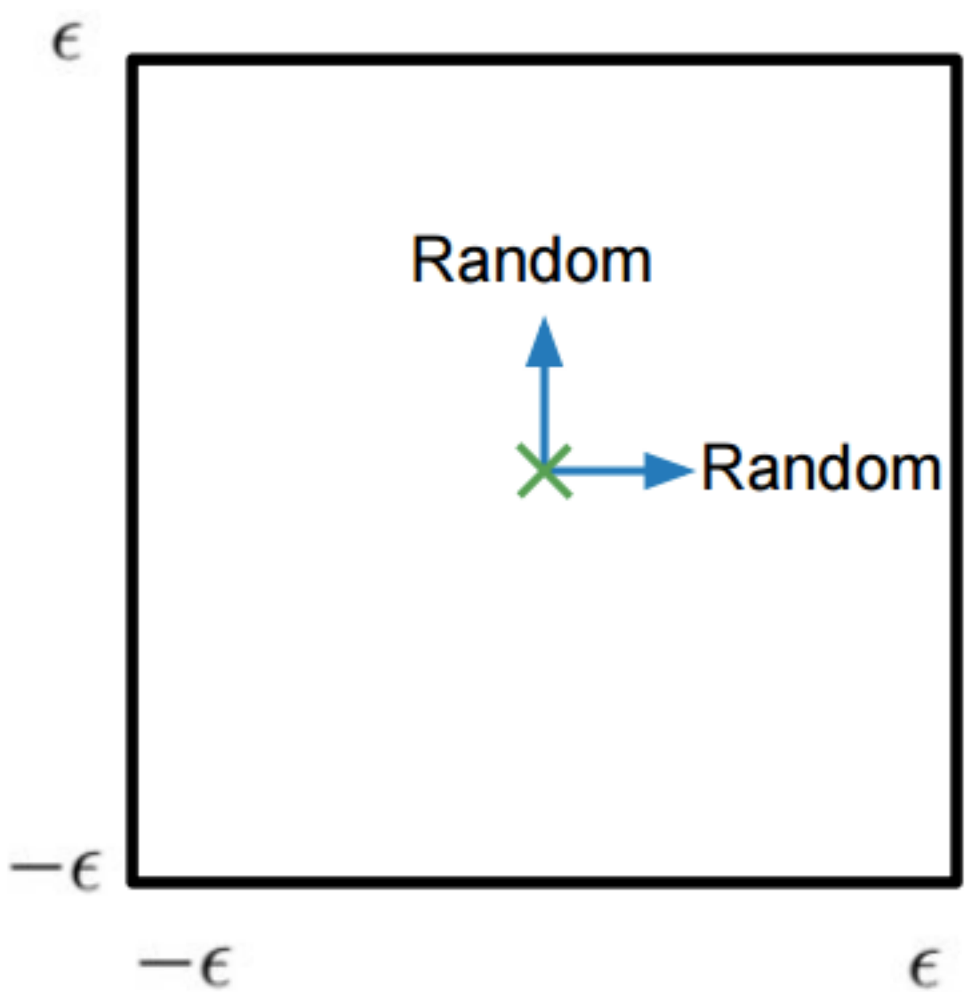
Maps of Adversarial Cross-Sections

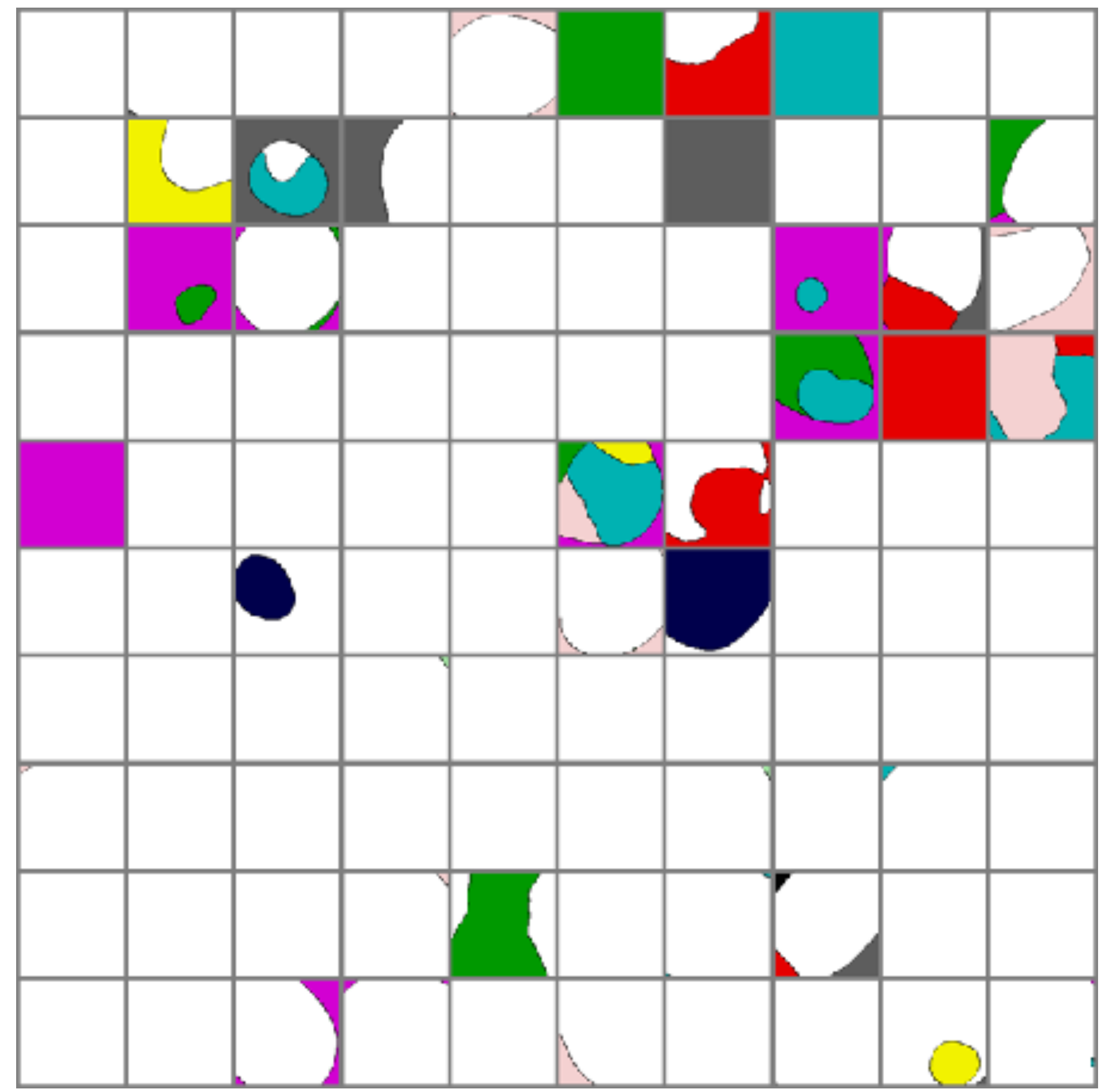


(Goodfellow 2016)

Maps of Random Cross-Sections

Adversarial examples
are not noise



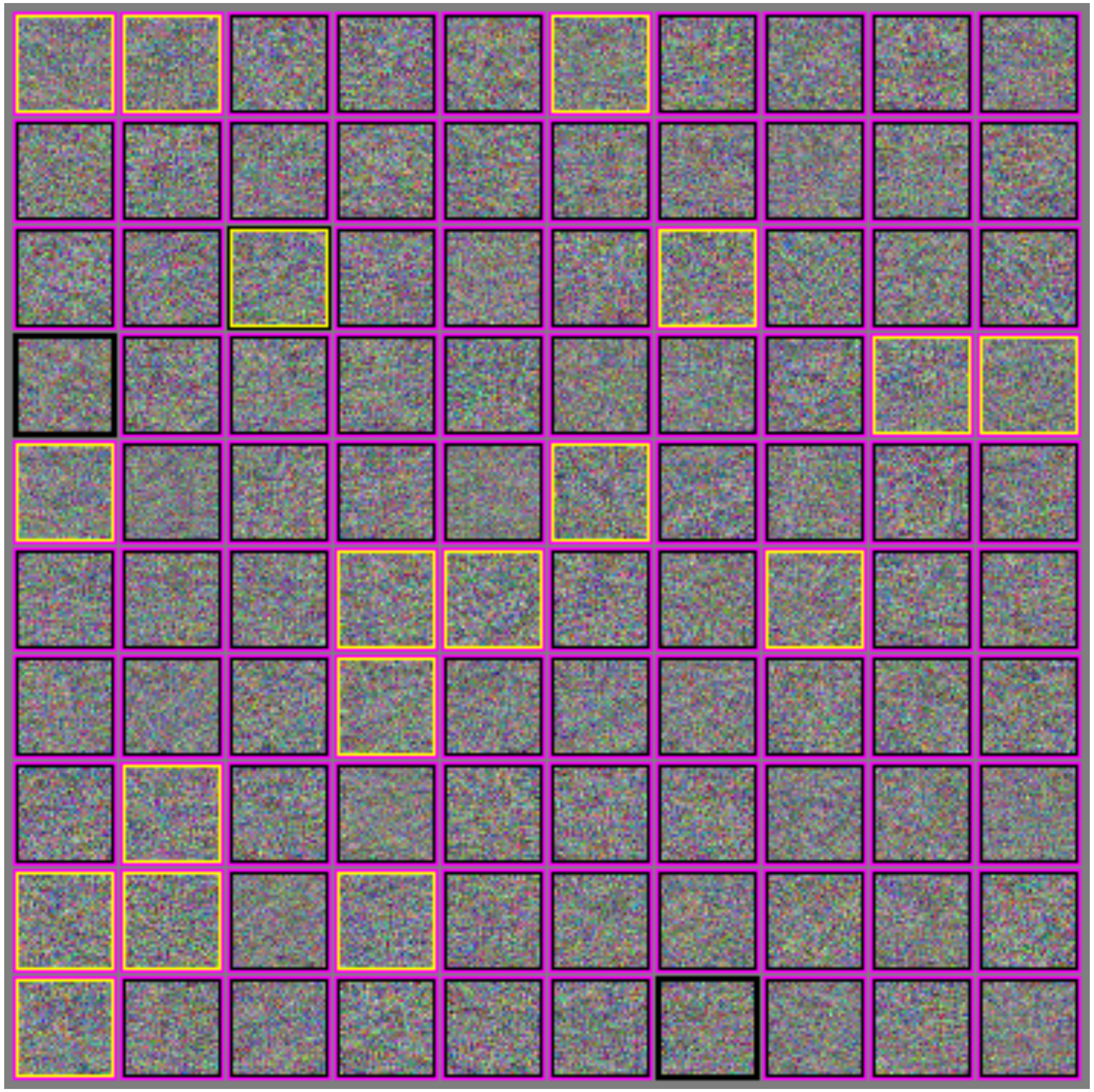


(collaboration with David Warde-Farley and Nicolas Papernot)

Clever Hans

("Clever Hans, Clever Algorithms," Bob Sturm)

Wrong almost everywhere



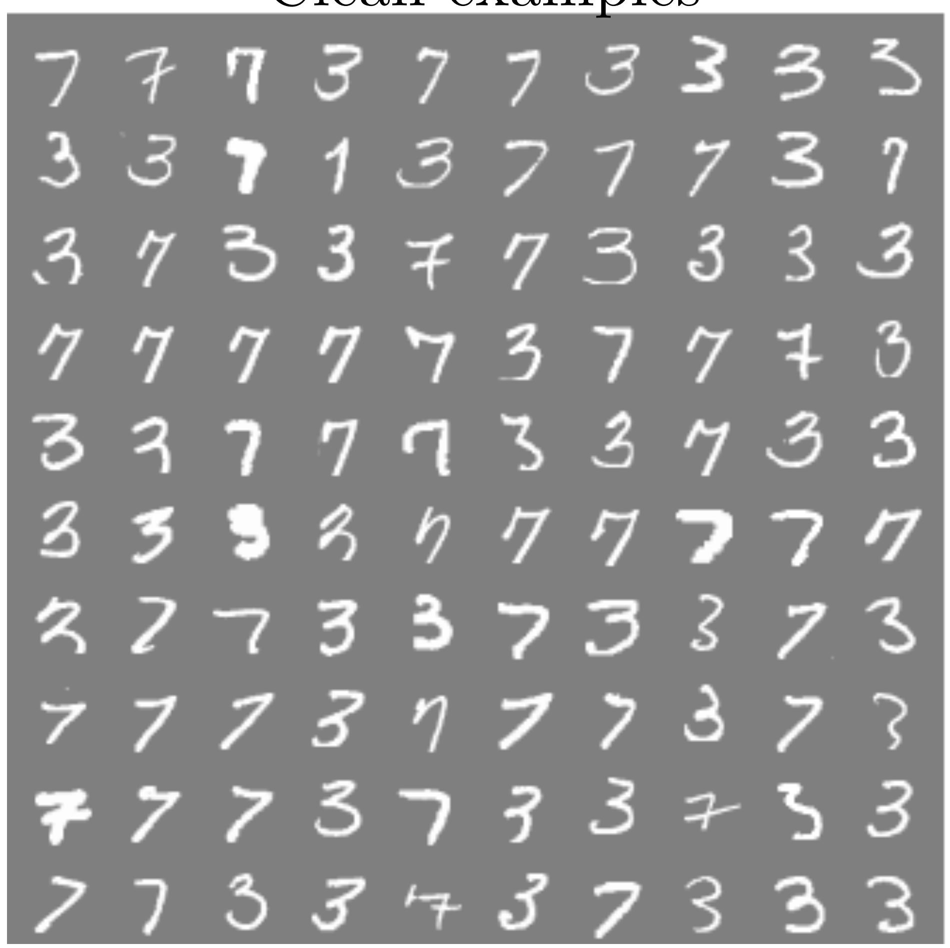
(Goodfellow 2016)

High-Dimensional Linear Models

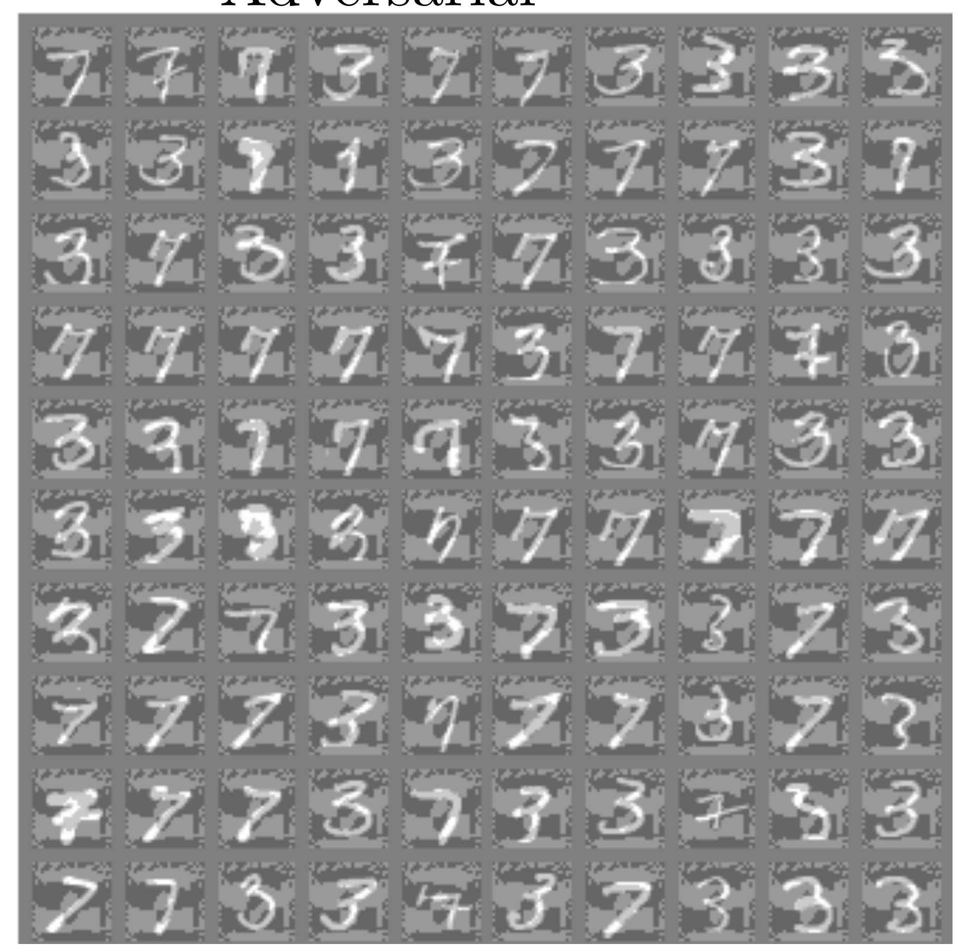
Weights

Signs of weights

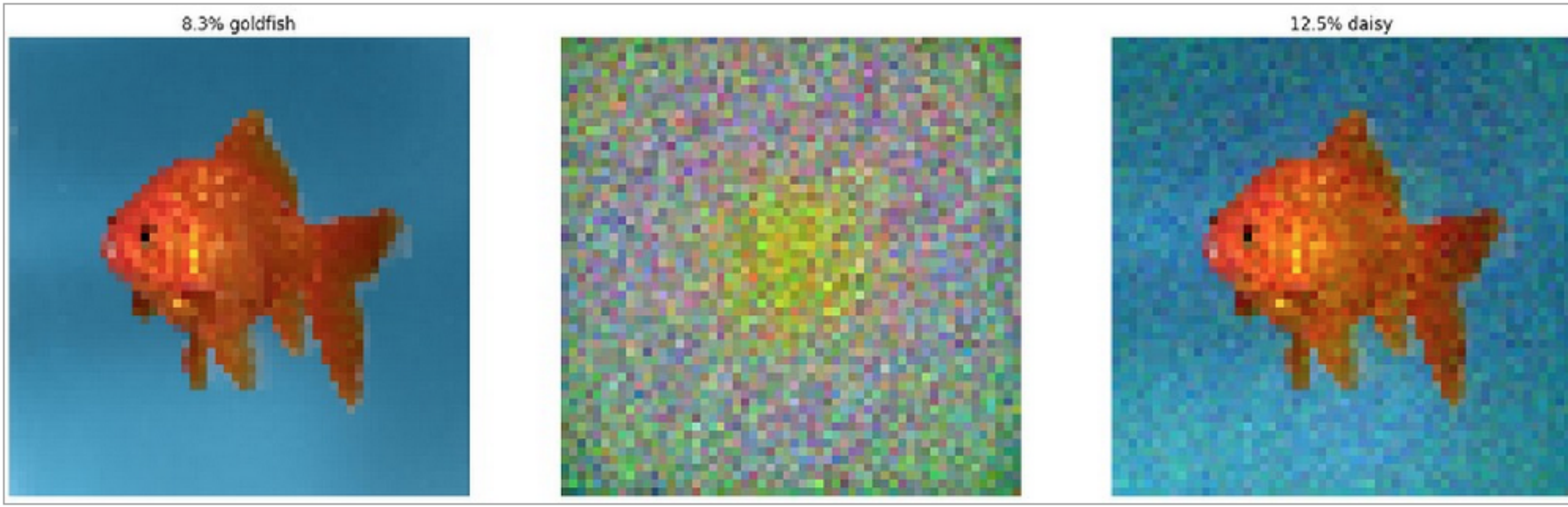
Clean examples



Adversarial

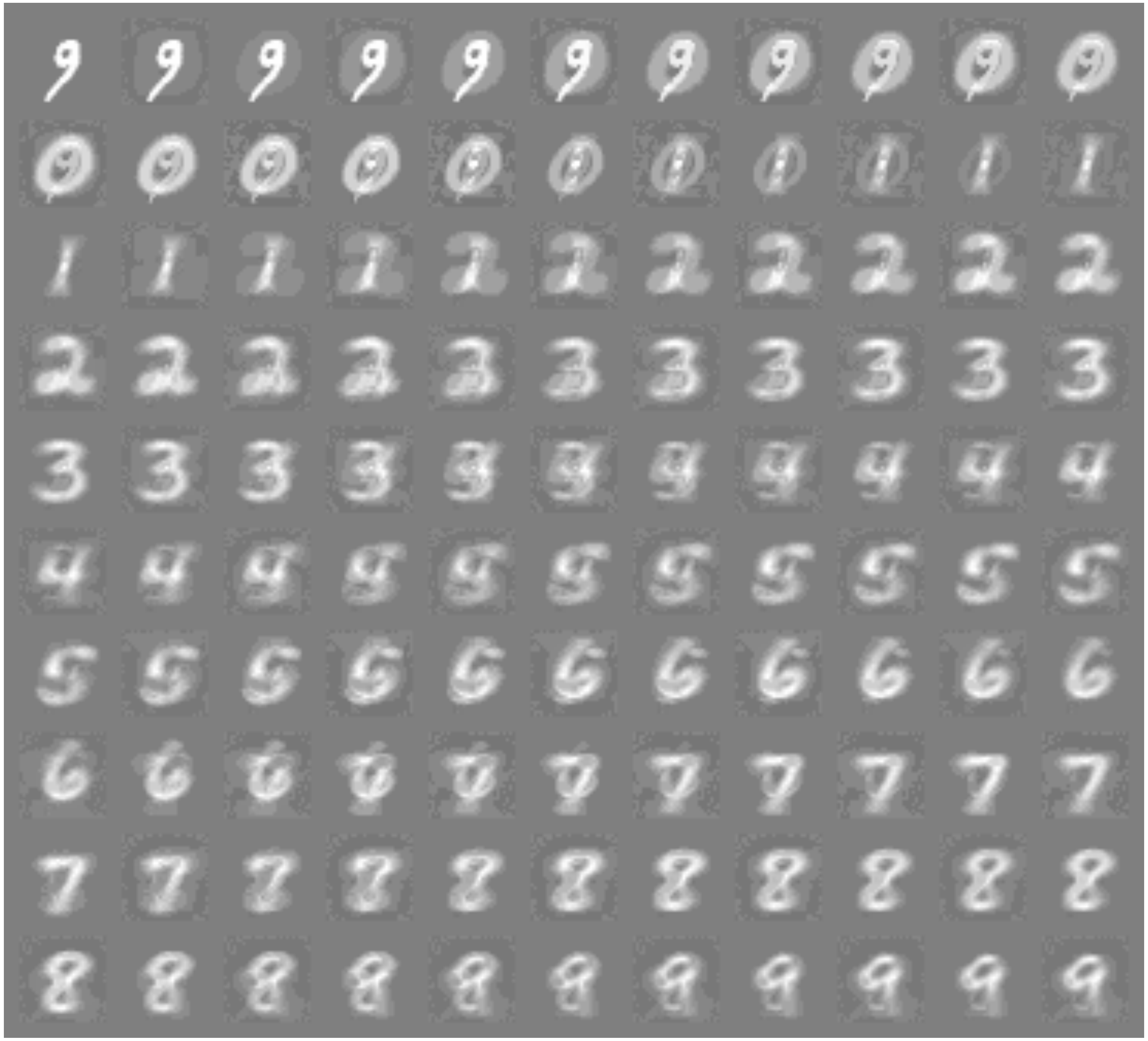


Linear Models of ImageNet



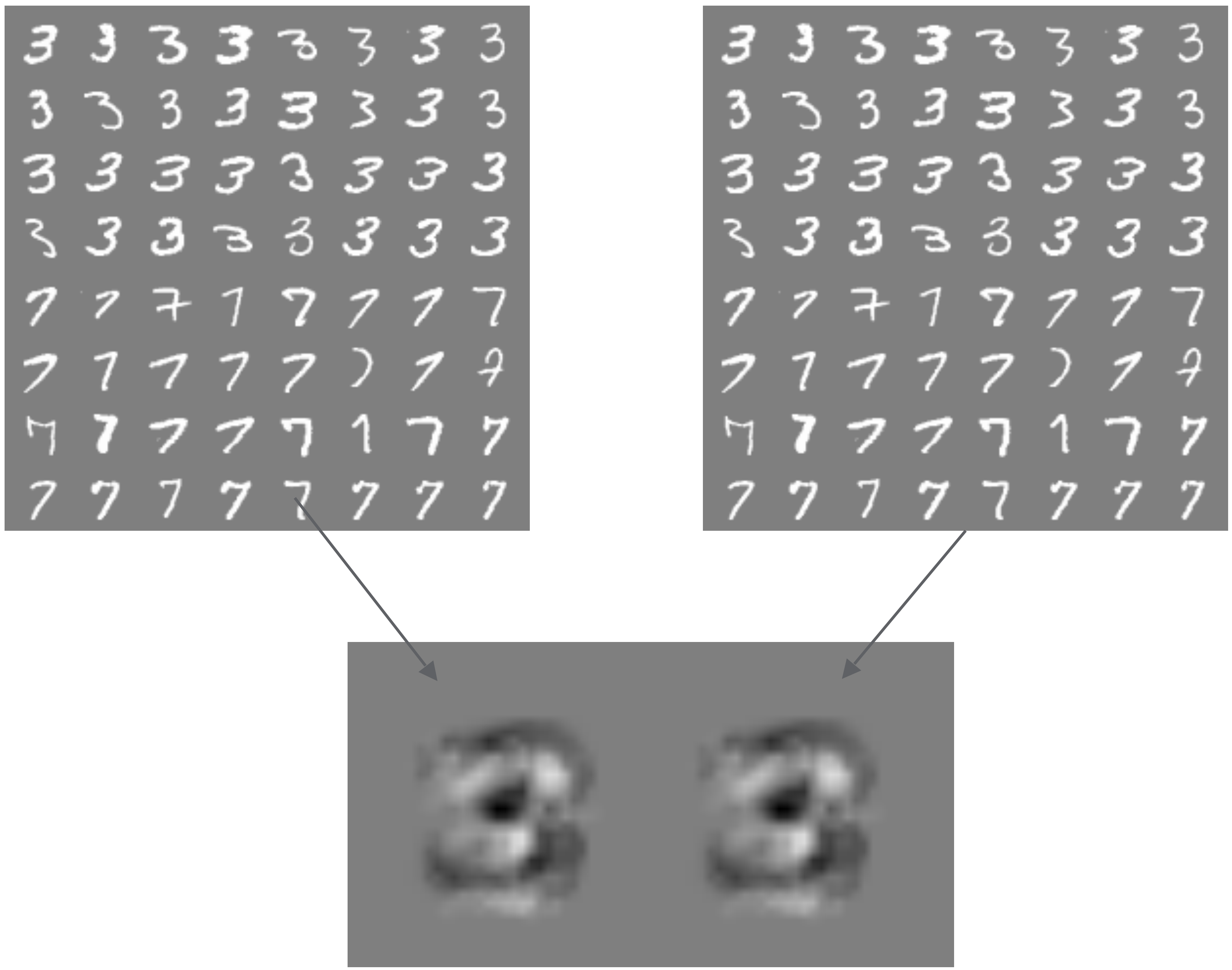
(Andrej Karpathy, "Breaking Linear Classifiers on ImageNet")

RBFs behave more intuitively



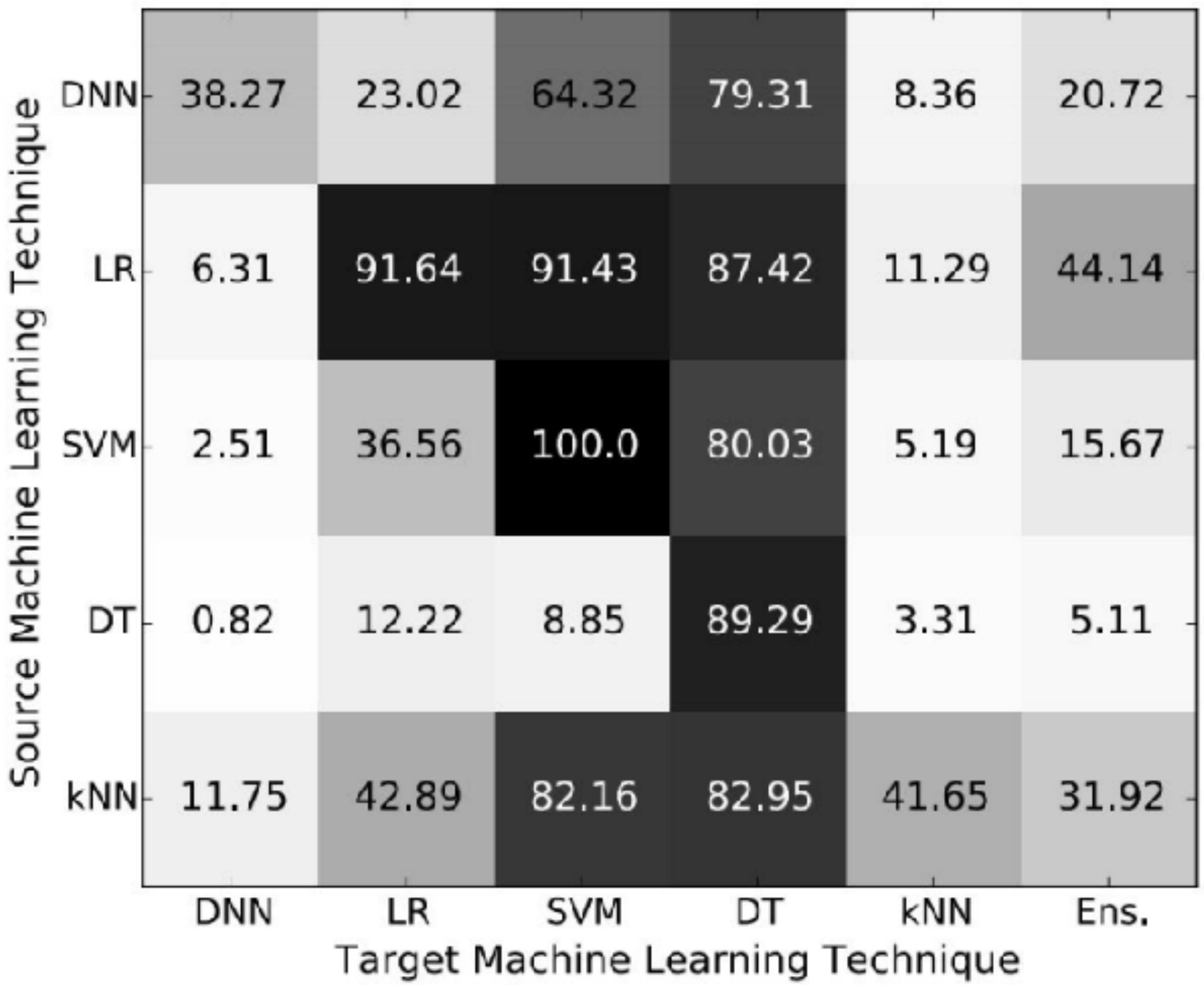
(Goodfellow 2016)

Cross-model, cross-dataset generalization



(Goodfellow 2016)

Cross-technique transferability



(Papernot 2016)

Transferability Attack

Target model with unknown weights, machine learning algorithm, training set; maybe non-differentiable

Train your own model

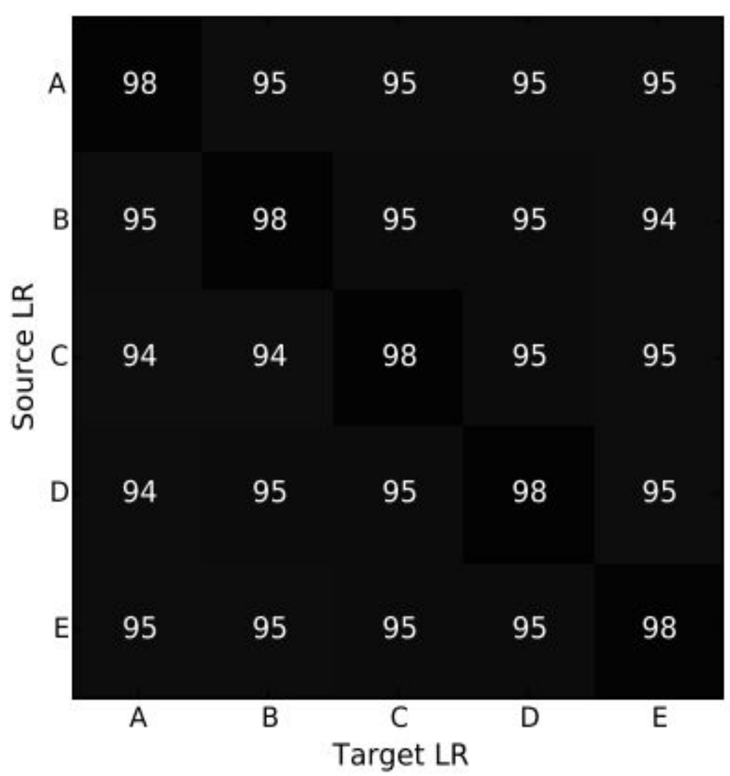
Substitute model mimicking target model with known, differentiable function

Adversarial crafting against substitute

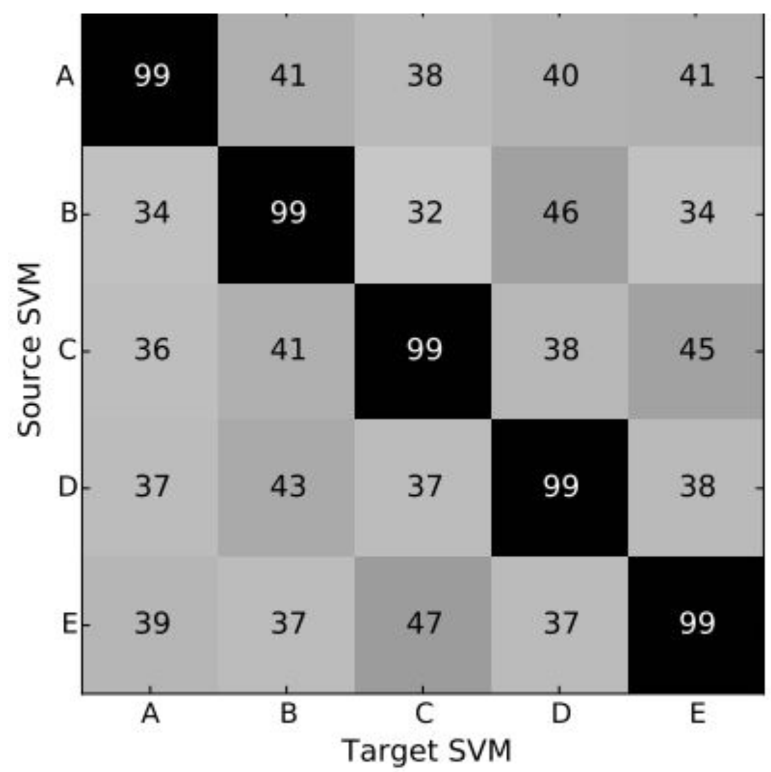
Adversarial examples

Deploy adversarial examples against the target; transferability property results in them succeeding

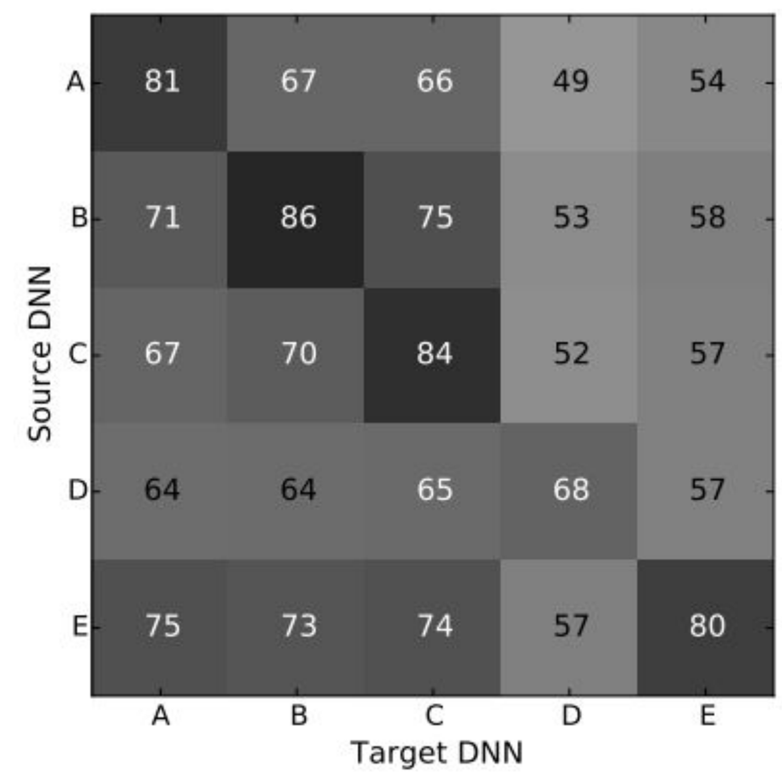
Cross-Training Data Transferability



Strong



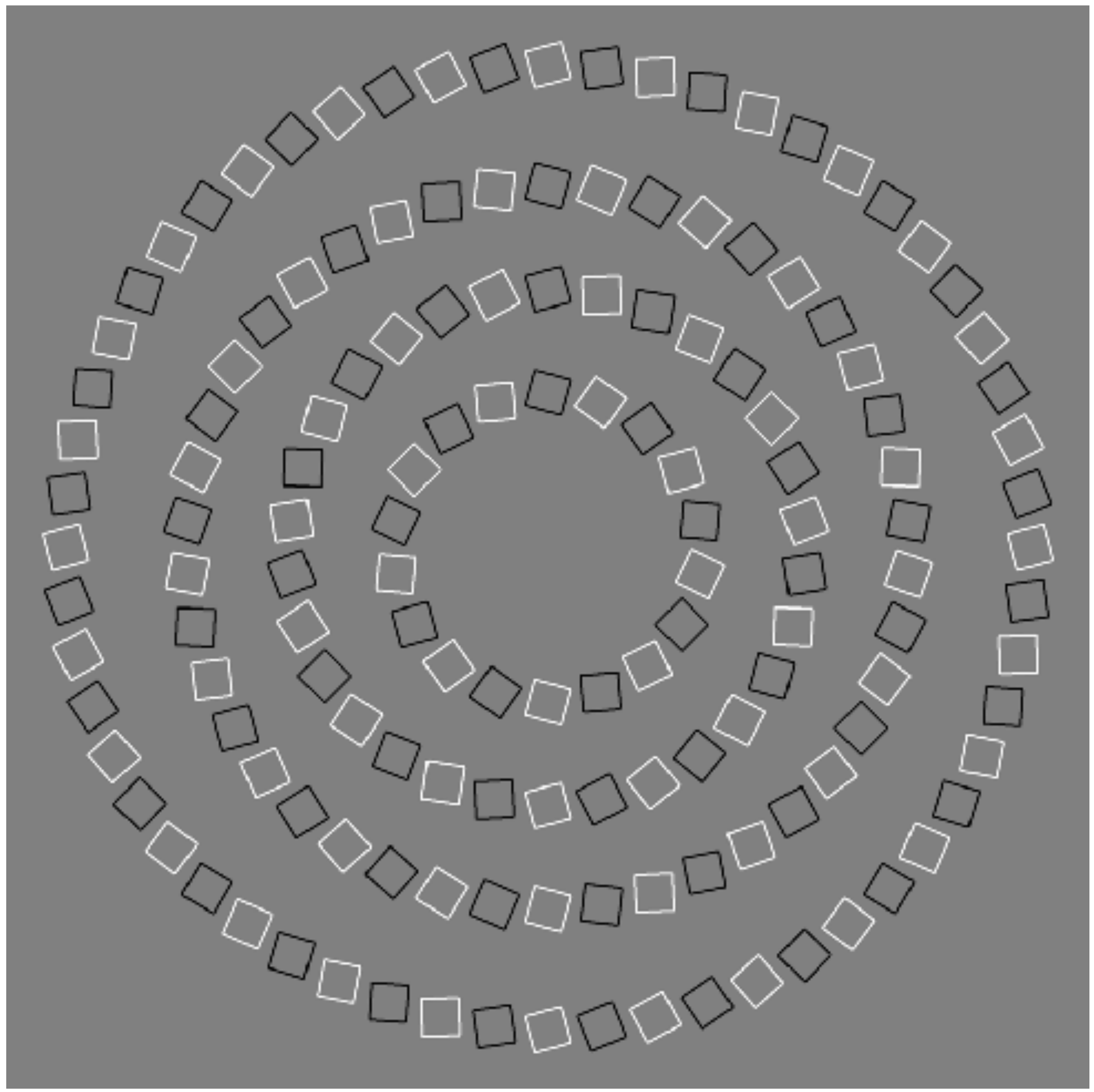
Weak



Intermediate

(Papernot 2016)

Adversarial Examples in the Human Brain



These are concentric circles, not intertwined spirals.

(Pinna and Gregory, 2002)

Practical Attacks

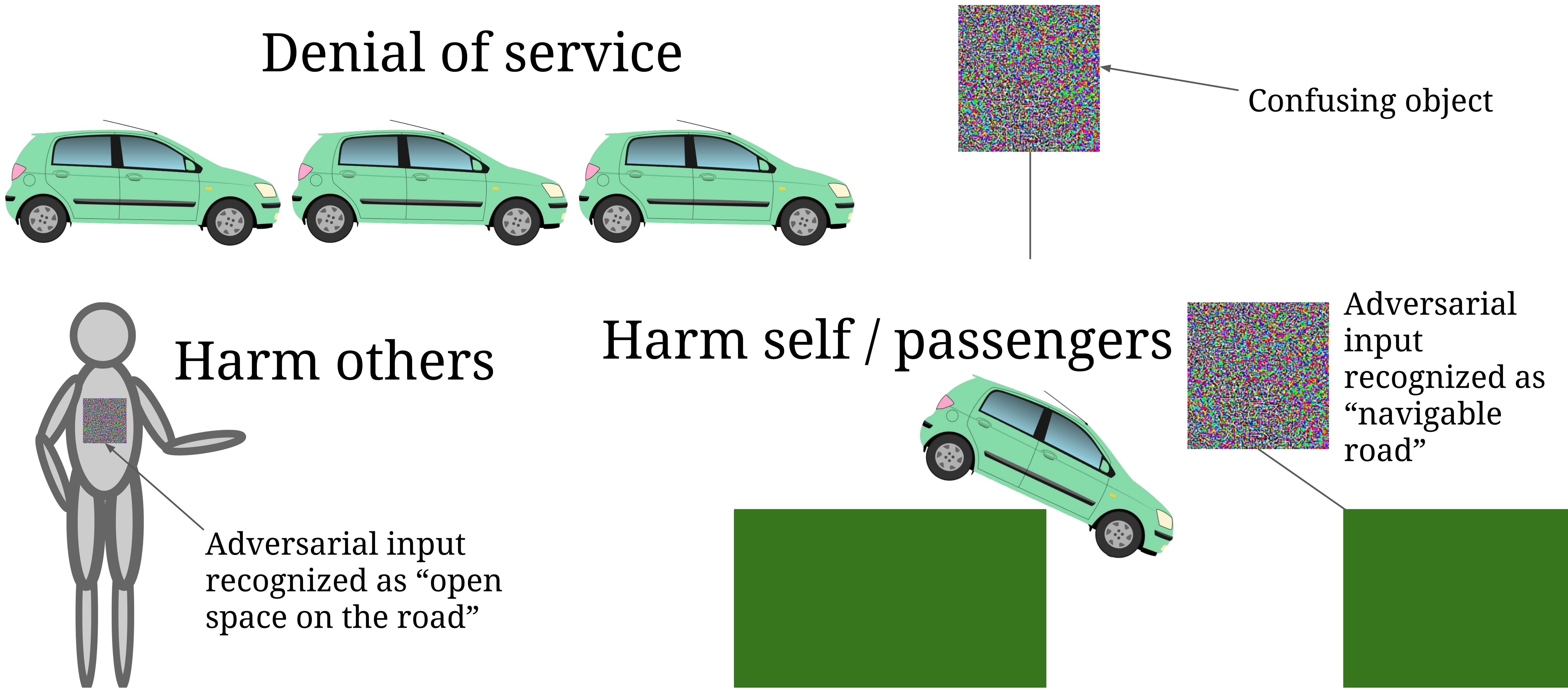
- Fool real classifiers trained by remotely hosted API (MetaMind, Amazon, Google)
- Fool malware detector networks
- Display adversarial examples in the physical world and fool machine learning systems that perceive them through a camera

Adversarial Examples in the Physical World

(a) Printout (b) Photo of printout (c) Cropped image

(Goodfellow 2016)

Hypothetical Attacks on Autonomous Vehicles



(Goodfellow 2016)

Failed defenses

Generative pretraining

Removing perturbation with an autoencoder

Adding noise at test time

Ensembles

Confidence-reducing perturbation at test time

Error correcting codes

Multiple glimpses

Weight decay

Double backprop

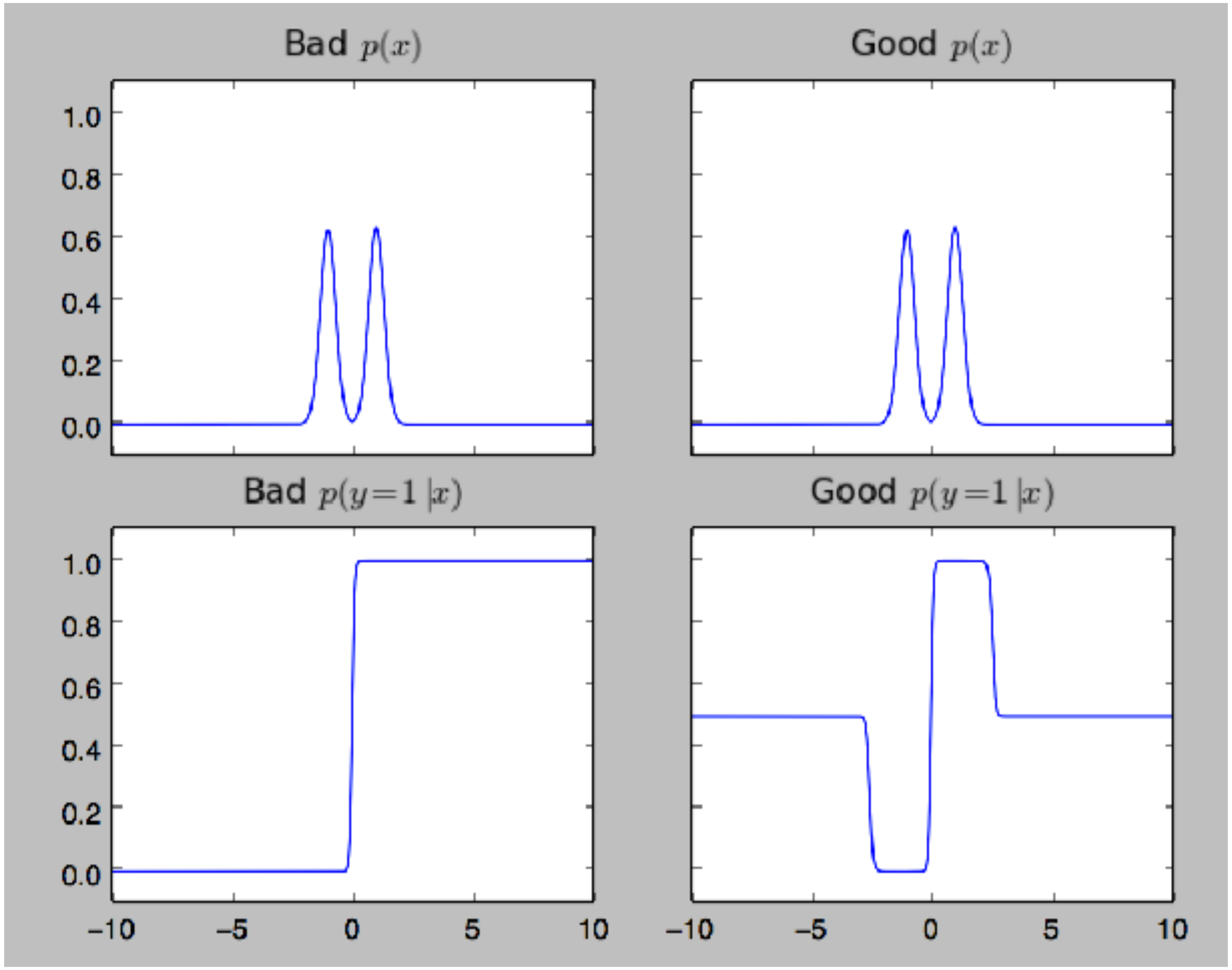
Adding noise at train time

Various non-linear units

Dropout

Generative Modeling is not Sufficient to Solve the Problem

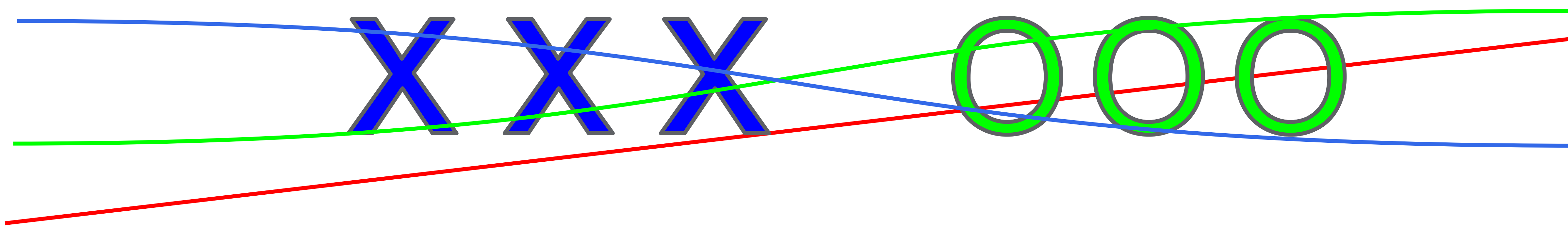
Both these two class mixture models implement roughly the same marginal over x, with very different posteriors over the classes. The likelihood criterion cannot strongly prefer one to the other, and in many cases will prefer the bad one.

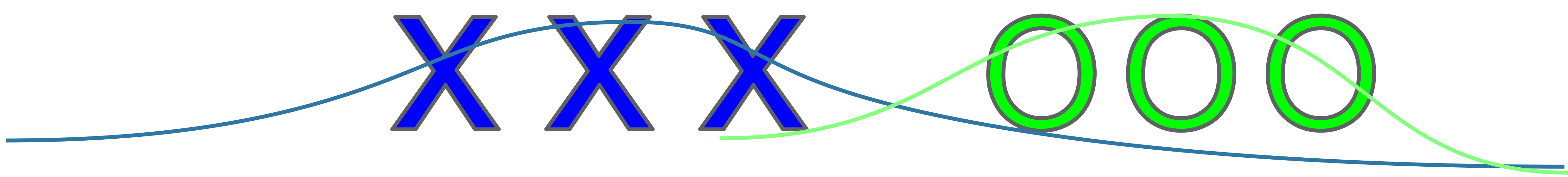


(Goodfellow 2016)

Universal approximator theorem

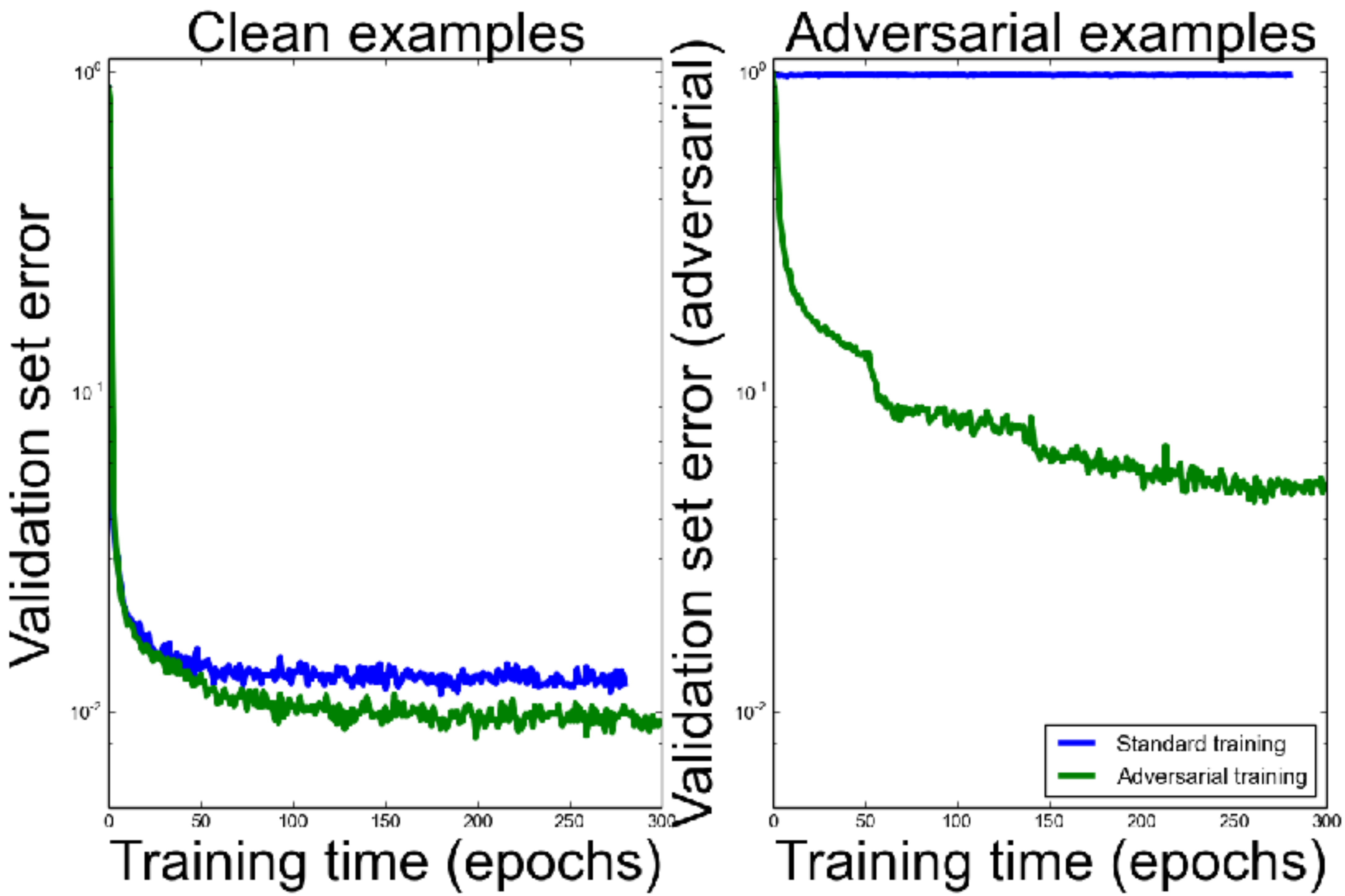
Neural nets can represent either function:





Maximum likelihood doesn't cause them to learn the right function. But we can fix that...

Training on Adversarial Examples



(Goodfellow 2016)

Adversarial Training of other Models

- Linear models: SVM / linear regression cannot learn a step function, so adversarial training is less useful, very similar to weight decay

- k-NN: adversarial training is prone to overfitting.

- Takeway: neural nets can actually become more secure than other models. *Adversarially trained neural nets have the best empirical success rate on adversarial examples of any machine learning model.*

Weaknesses Persist

(Goodfellow 2016)

Adversarial Training

Labeled as bird

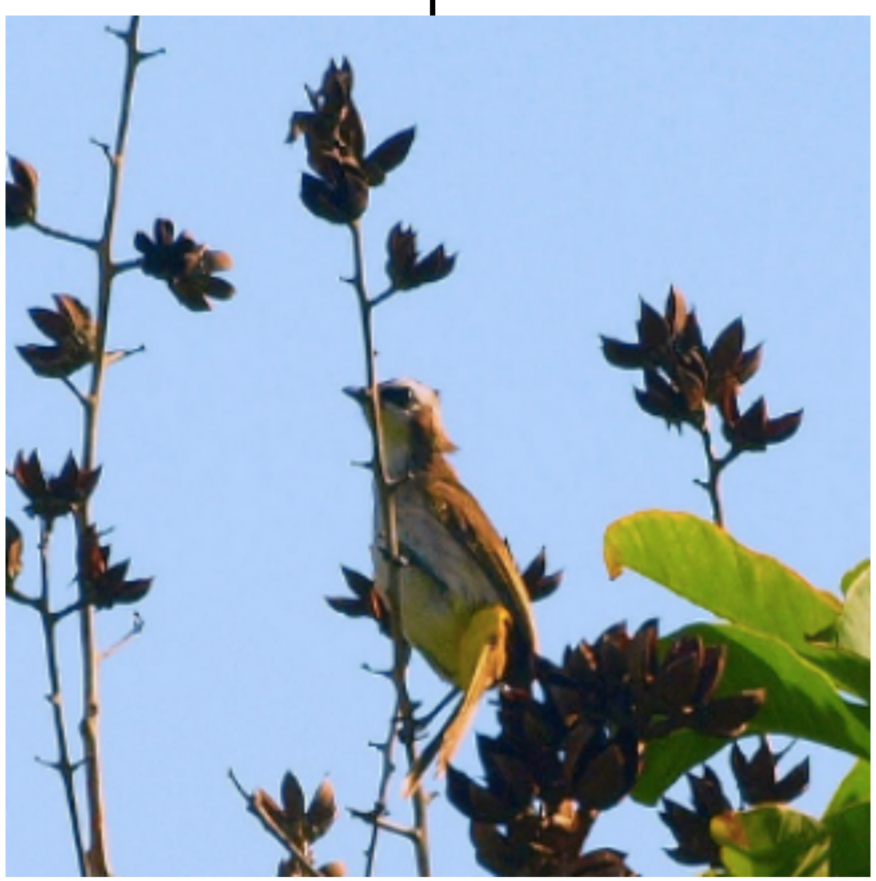
Still has same label (bird)

Decrease probability of bird class

Virtual Adversarial Training

Unlabeled; model guesses it's probably a bird, maybe a plane

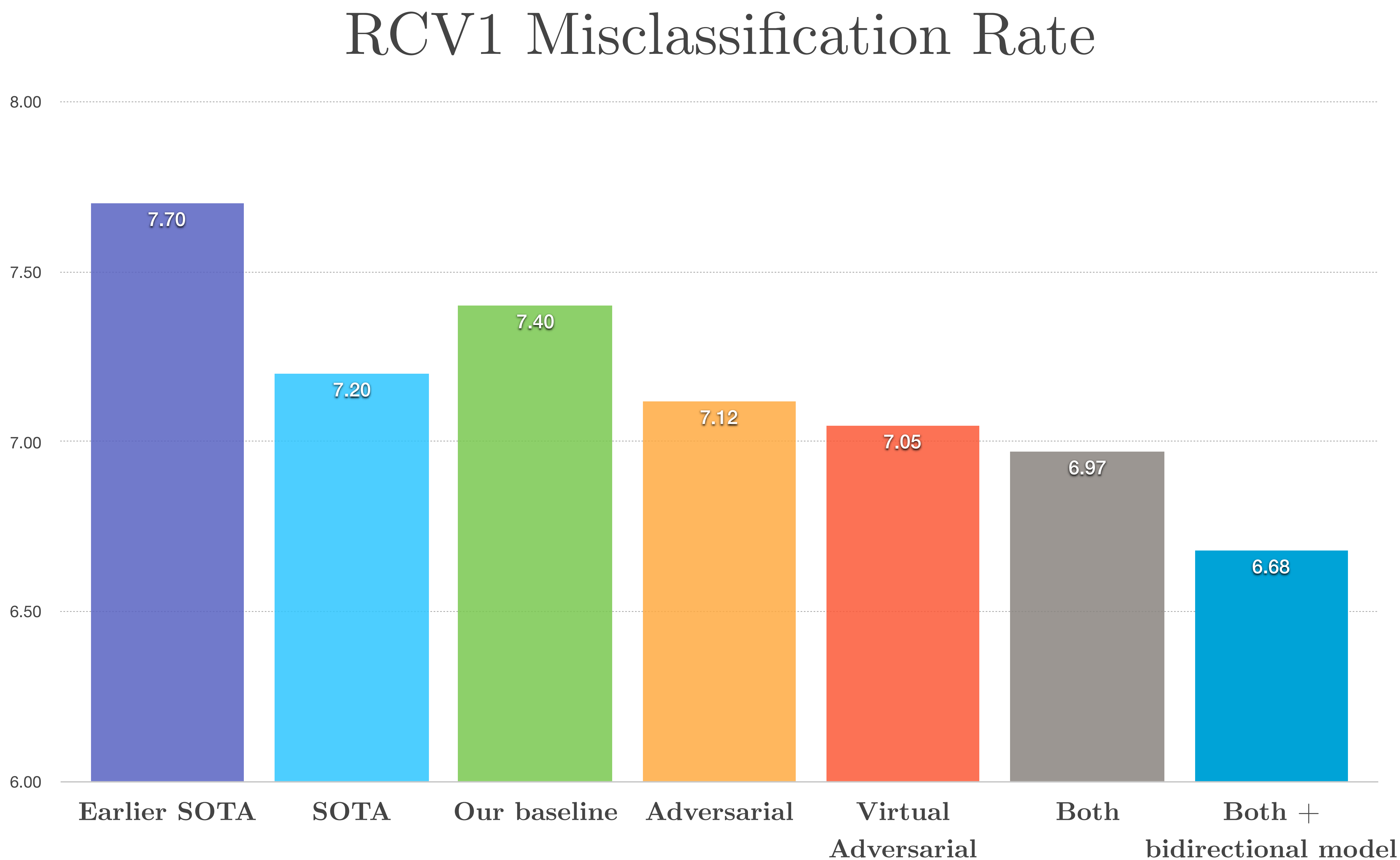
New guess should match old guess (probably bird, maybe plane)



Adversarial perturbation intended to change the guess

(Goodfellow 2016)

Text Classification with VAT



Zoomed in for legibility

(Goodfellow 2016)

Universal engineering machine (model-based optimization)

Make new inventions by finding input that maximizes model's predicted performance

Training data

Extrapolation

(Goodfellow 2016)

Conclusion

- Attacking is easy
- Defending is difficult
- Benchmarking vulnerability is training
- Adversarial training provides regularization and semi-supervised learning
- The out-of-domain input problem is a bottleneck for model-based optimization generally

cleverhans

Open-source library available at:

https://github.com/openai/cleverhans

Built on top of TensorFlow (Theano support anticipated)

Standard implementation of attacks, for adversarial training and reproducible benchmarks