EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES

Nazmul Karim, Umar Khalid

Overview

- ❑ Introduction.
- ❑Previous Findings.
- ❑ Motivation.
- ❑ Proposed Adversarial Attack Method.
- ❑ Effect of Adversarial Training.
- ❑ Generalization of Adversarial Attacks.
- ❑ Strengths & Weaknesses.

Paper History and Authors

- Motivated by the findings from "Intriguing Properties of Neural Networks" paper.
- Accepted at ICLR 2015 conference.

- Ian Goodfellow
  - Inventor of Generative Adversarial Network (GAN)
  - Citations 120k+
- Jonathan Shlens
  - Citations 63k+
- Christian Szegedy
  - Citations 100k+

Previous Findings

- Szegedy et al. (2014) demonstrated some properties of neural network:

I. The differences between original samples and adversarial examples were indistinguishable

II. Adversarial examples are transferrable.

III. Models with different architectures trained on different subsets may misclassify

IV. Training on adversarial examples can regularize the model

Motivation

Causes of adversarial examples a mystery:

Extreme non-linearity of NNs?

Insufficient model averaging?

Insufficient regularization?

Models are not learning the true underlying properties of the data

Potential of adversarial examples for use in adversarial training?

Adversarial Examples for Linear Models

- First, consider a linear model.
- With 8 bits for pixel value, a pixel can have values of 0-255 at most.
- We can perturb each pixel of the image.
- If a perturbation η is added to an image x, that will give us,

$$\tilde{x} = x + \eta,$$

$$\|\eta\|_\infty < \varepsilon$$

- Dot product of weight vector and adversarial example:

$$w^T \tilde{x} = w^T x + w^T \eta$$

Adversarial Examples for Linear Models

- The adversarial perturbation causes the activation to grow by: $w^T \eta$
- Maximum of this growth subject to the max norm constraint on η by assigning:

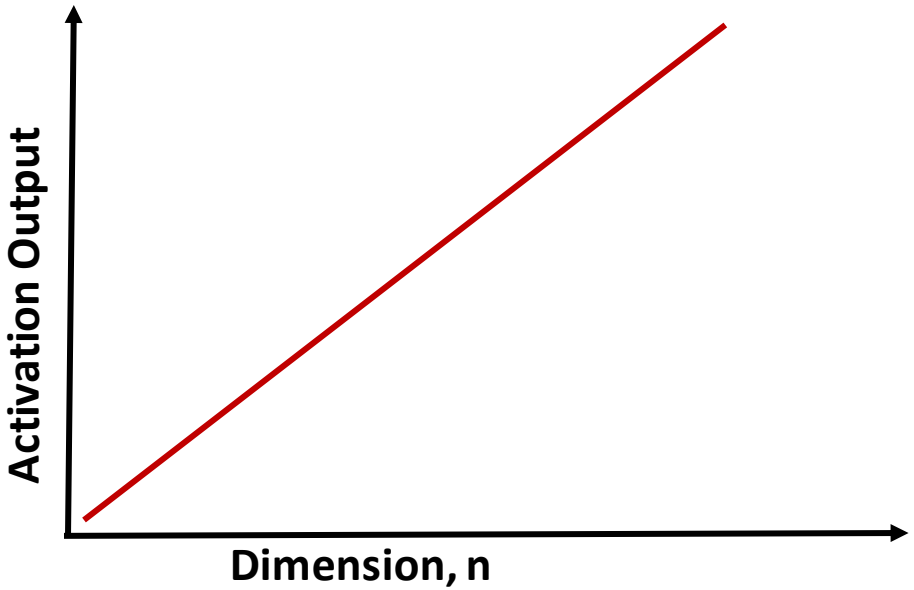
$$\eta = \text{sign}(\boldsymbol{w})$$

- Activation will grow by ϵmn

n= dimensions of W
m=average magnitude of an element of the weight vector

Adversarial Examples for Linear Models

- If input x has sufficient dimensionality (n), small perturbation could make huge output change.



Perturbation. : Constant
Activation output: Grows with dimension

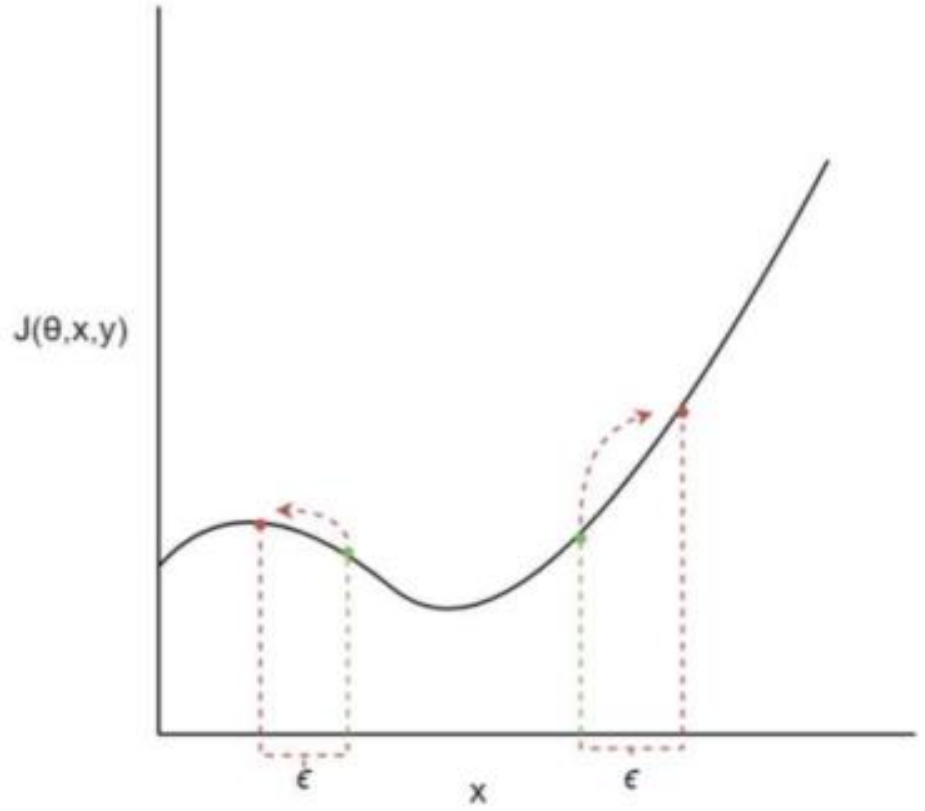
Summary: A simple linear model can have adversarial examples if the input has sufficient dimensionality.

Adversarial Example for Non-Linear Models

- Is it applicable for nonlinear models?
- Linear behavior of Neural Networks.
- We can generate perturbation using Fast Gradient Sign Method (FGSM),

$$\boldsymbol{\eta} = \epsilon \operatorname{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$$

Θ: be the parameters of a model
x: The input to the model.
y**:** The target associated with x.
***J* :** Cost function.



Adversarial Example for Non-Linear Models

x

"panda"
57.7% confidence

$+ .007 \times$

$\text{sign}(\nabla_{\boldsymbol{x}} J(\theta, x, y))$

"nematode"
8.2% confidence

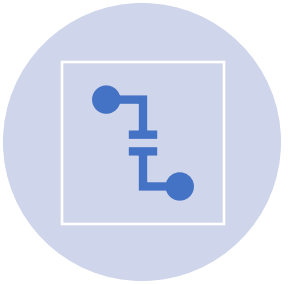
$=$

$x + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\theta, x, y))$

"gibbon"
99.3 % confidence

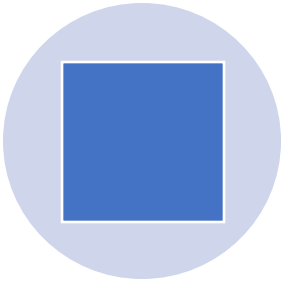
Summarizing FGSM

Single-step, **untargeted, white-box attack.**



Exploits the linearity assumption of a model.

Changes every pixel in x to follow gradient .



Fast **to compute, effective in fooling models.**

Experimental Results

- FGSM based attack on Neural networks with different activation function.
- Experiment with two datasets.

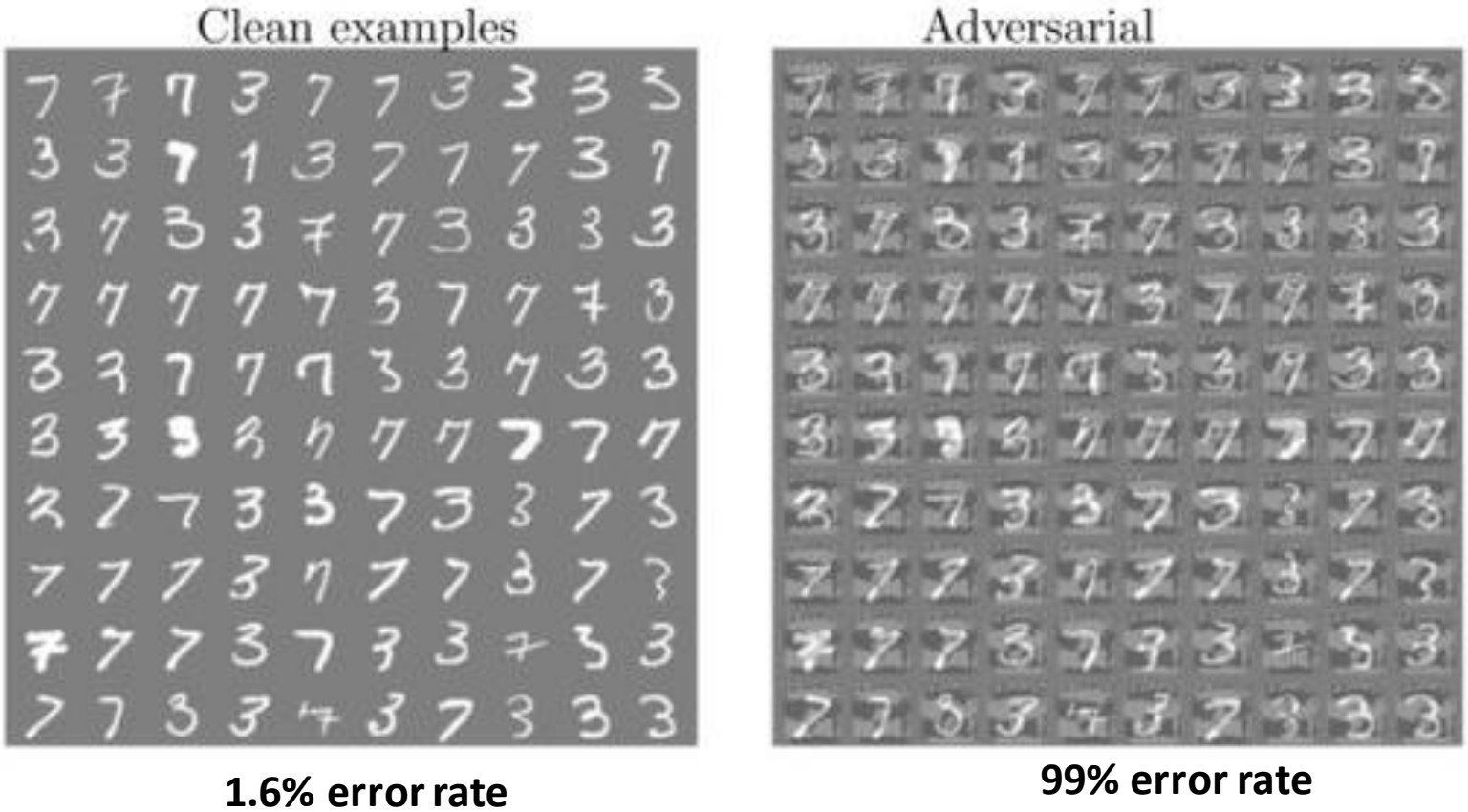
ε	Error Rate	Confidence	Activation	Dataset
0.25	99.9%	79.3%	Sigmoid	MNIST
0.25	89.4%	97.6%	Maxout	MNIST
0.1	87.15%	96.6%	Maxout	CIFAR10

Adversarial Training (AT)

- Training on adversarial examples
- Assign an adversarial example the same label as the data point it is close to.
- A way of minimizing the worst-case error.


Adversarial Training
for Linear Models

FGSM Attack to a Logistic Regression Model

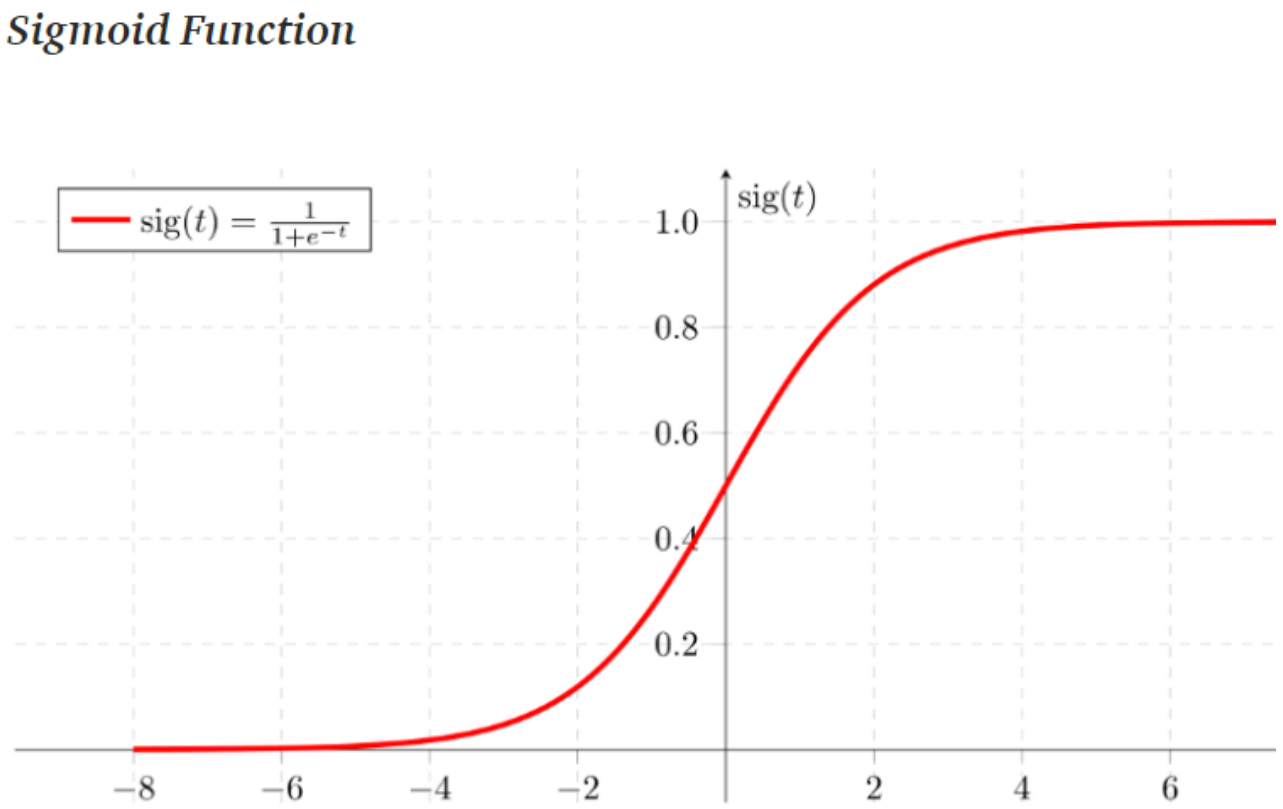


1.6% error rate

99% error rate

Logistic Regression Model

- $P(y = 1) = \sigma(w^T x +$
  b) where σ(z) is the logistic sigmoid function



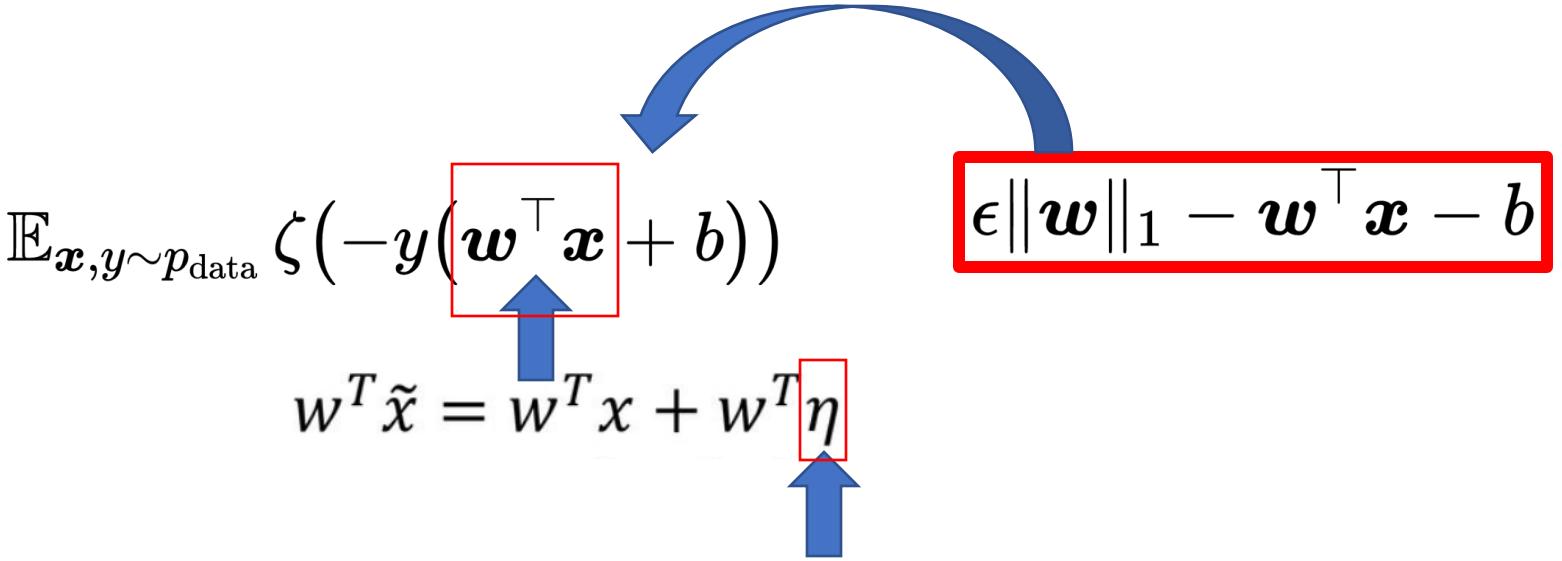
Logistic Regression Model

- Training consists of gradient descent on:

$$\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}}\,\zeta\big(-y\big(\boldsymbol{w}^{\top}\boldsymbol{x}+b\big)\big)$$

$$\zeta(z)=\log(1+\exp(z))$$ *Softplus Function*

Adversarial Training for Logistic Regression Model



$$\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}}\,\zeta\left(-y\left(\boldsymbol{w}^{\top}\boldsymbol{x}+b\right)\right) \qquad \epsilon\|\boldsymbol{w}\|_1-\boldsymbol{w}^{\top}\boldsymbol{x}-b$$

$$w^T\tilde{x}=w^Tx+w^T\eta$$

$$\boldsymbol{\eta}=\epsilon\,\text{sign}(\nabla_{\boldsymbol{x}}J(\boldsymbol{\theta},\boldsymbol{x},y))$$

$$w^T\eta \;=\; -\;\epsilon\|\boldsymbol{w}\|_1$$

Where ***sign of Gradient = -sign(W)***, and

$$\boldsymbol{w}^{\top}\,\text{sign}(\boldsymbol{w})=\|\boldsymbol{w}\|_1$$

Adversarial Training for Logistic Regression Model

The adversarial version of logistic regression is therefore to minimize :

$$\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}}\,\zeta\big(y\big(\epsilon\|\boldsymbol{w}\|_1-\boldsymbol{w}^\top\boldsymbol{x}-b\big)\big)$$

Is it similar to L1 regularization?

L1 regularization for Logistic Regression Model

- To prevent the overfitting problem
- Adds a regularization term
- **Minimize both the Loss term and the regularization term**.
- **λ** = 0, it will overfit
- large **λ** leads to underfit.

$$\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}}\,\zeta\big(-y\big(\boldsymbol{w}^{\top}\boldsymbol{x}+b\big)\big) + \boldsymbol{\lambda}^{*}\,||\mathbf{w}||\mathbf{1}$$

$$\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}}\,\zeta\big(y\big(\epsilon\|\boldsymbol{w}\|_1-\boldsymbol{w}^{\top}\boldsymbol{x}-b\big)\big)$$

Adversarial Training vs L1 weight decay

- Training maxout networks on MNIST
- Good results using adversarial training with $\varepsilon = 0.25$
- L1 weight decay to the first layer
- Coefficient of .0025 caused 5% error on the training set

Adversarial Training of DNN

- DNN can become more resistant to adversarial examples.

- Cost Function for adversarial training,

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1 - \alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \operatorname{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)))$$

Adversarial Training of DNN

- Train the DNN on a training set of clean and adversarial examples.
- **Benefit:** It regularizes the model.

ERROR RATE ON MNIST TRAINING SET

Without Adversarial Training	With Adversarial Training
0.94%	0.84%

Adversarial Trained Model

- More robust to adversarial examples.
  - The model misclassified adversarial examples:
    - **Without training, 89.4%**
    - **With training, 17.9%.**

- Still misclassification.
  - Average confidence of misclassified examples: **81.4%.**

Other Considerations

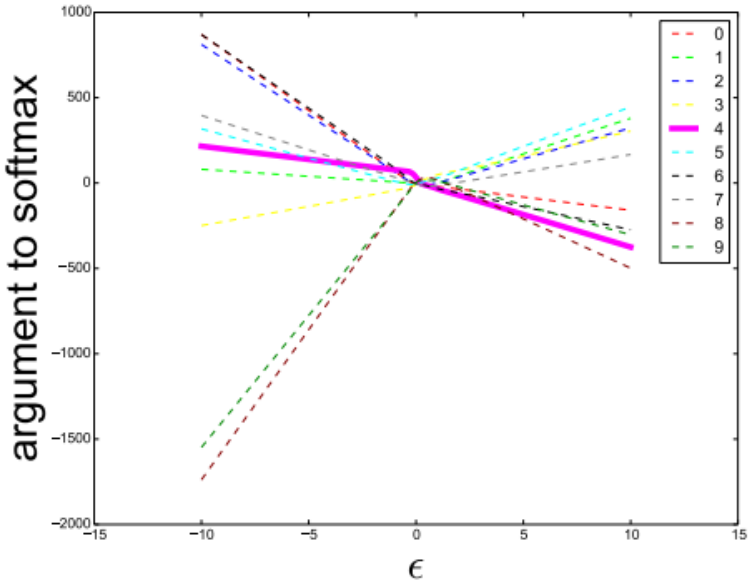
- What should we perturb?
  - The input
  - Hidden layer
  - Both
- Findings
  - The results are inconsistent for different settings.
  - Adding perturbation to hidden layer does not offer strong regularization as the input.

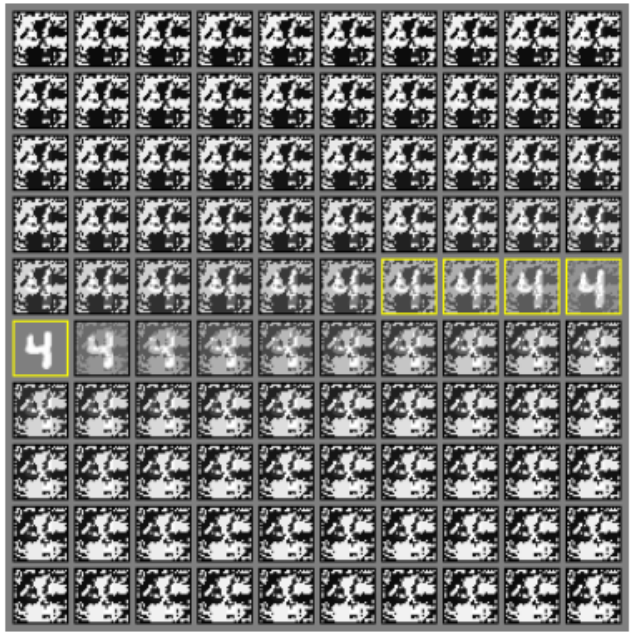
Why Do Adversarial Examples Generalize ?

- Different models misclassify the same adversarial example?
- Extreme non-linearity and overfitting cannot readily account for this behavior
- Adversarial examples occur in broad subspaces
- Adversarial examples are abundant

Why Do Adversarial Examples Generalize ?

- Adversarial examples occur reliably for almost any sufficiently large value of ϵ.
- Wrong classifications are stable across a wide region of values





Trained maxout network for MNIST 10 classes

Generalization of Adversarial Examples

- The neural networks trained with current methodologies all resemble the linear classifier
- The classifier learns approximately the same classification weights when trained on different subsets of the training set

Consistent with linear behavior being a major cause of cross-model generalization.

Alternative Hypothesis

- Generative training could provide more constraint on the training process
  - ✓The MP-DBM (Goodfellow et al., 2013a) provides a good model to test this hypothesis

RESULT?

The mere fact of being generative is not alone sufficient

Alternative Hypothesis

- Averaging over many models can cause adversarial examples to wash out

  ✓Trained an ensemble of twelve maxout networks on MNIST
  ✓Error rate of 91.1% on adversarial examples

RESULT?

Ensembling provides only limited resistance to adversarial perturbation.

Strengths

- Explained the probable reason for the existence of adversarial examples.
- Proposed a simple and effective way of generating adversarial examples.
- Does not use any iterative method which is computationally efficient.
- Presented their case for both linear and non-linear models.
- FGSM based adversarial training creates robust models.
- Adversarial examples are generalizable

Weaknesses

- Effect of different type of regularization techniques needs to be studied.
- Limited study on Model Capacity/ Depth.
- Limited experiments to prove the hypothesis of generalization of adversarial attacks.
- Not concrete explanation of attack transferability.
- Ensembling has been refuted but with little evidence.

Summary

- Adversarial generation is due to linear property of high dimensional dot products.
- The generalization is due to similar learnt weights
- Linear models failed to resist the attack.
- Adversarial training can result in regularization
- Ensembles are not resistant to adversarial examples.
- Generative technique is not sufficient

Questions