Adversarial examples in deep learning

G. Châtel

06/07/2017

1. Introduction

2. Attack

3. Defense

Basic notions

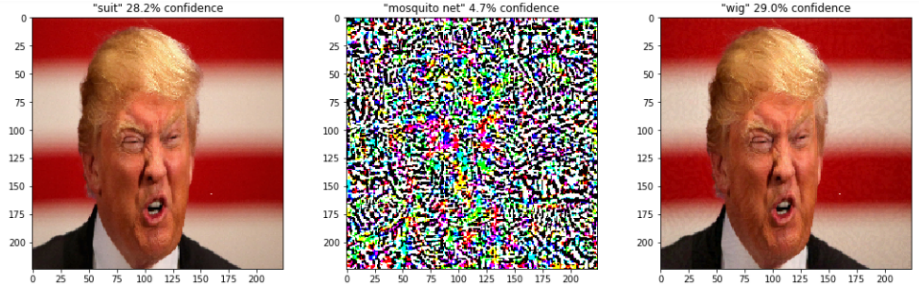
Adversarial example

An *adversarial example* is a sample of input data which has been modified *very slightly* in a way that is intended to cause a machine learning classifier to misclassify it.

Basic notions

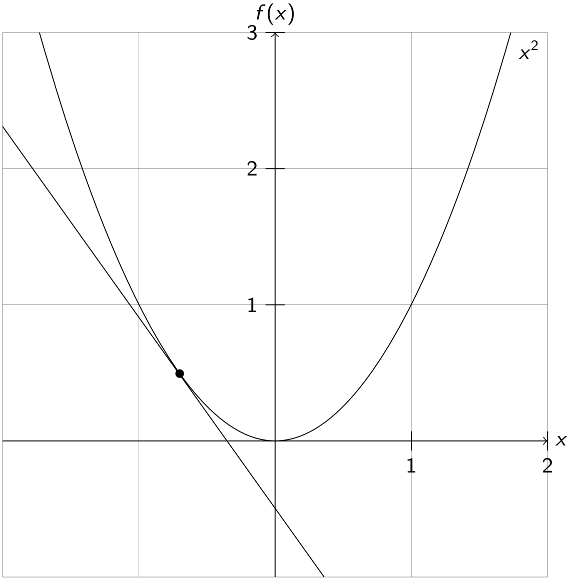
Adversarial example

An *adversarial example* is a sample of input data which has been modified *very slightly* in a way that is intended to cause a machine learning classifier to misclassify it.



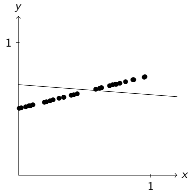
Gradient descent

Basic concept



Gradient descent

Model optimization

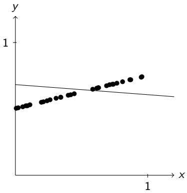


We have a set of points that we want to approximate with a line.

$$y = ax + b$$

Gradient descent

Model optimization



We have a set of points that we want to approximate with a line.

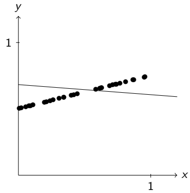
$$y = ax + b$$

First we choose a loss that measures how good our predictions are.

$$l(x, y, a, b) = (y - (ax + b))^2$$

Gradient descent

Model optimization



We have a set of points that we want to approximate with a line.

$$y = ax + b$$

First we choose a loss that measures how good our predictions are.

$$l(x, y, a, b) = (y - (ax + b))^2$$

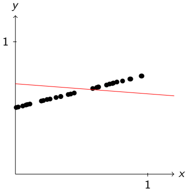
We compute how the loss is affected by small changes of a and b:

$$\frac{\mathrm{d}l}{\mathrm{d}a} = 2x(ax + b - y) \qquad \frac{\mathrm{d}l}{\mathrm{d}b} = 2(ax + b - y)$$

And we update a and b iteratively until we reach a satisfying result (the average loss is low enough).

Gradient descent

Being evil

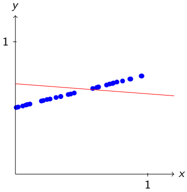


In our previous example, we have modified the model in order to minimize the loss.

$$y = ax + b$$

Gradient descent

Being evil



In our previous example, we have modified the model in order to minimize the loss.

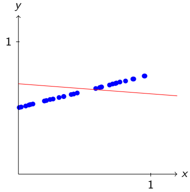
$$y = ax + b$$

Now suppose we are an attacker who wants to maximise the loss of a model, its parameters being fixed. The only thing we can modify is the inputs.

$$l(x, y, a, b) = (y - (ax + b))^2$$

Gradient descent

Being evil



In our previous example, we have modified the model in order to minimize the loss.

$$y = ax + b$$

Now suppose we are an attacker who wants to maximise the loss of a model, its parameters being fixed. The only thing we can modify is the inputs.

$$l(x, y, a, b) = (y - (ax + b))^2$$

In order to do this, we compute how the loss is affected by small changes of the input:

$$\frac{\mathrm{d}l}{\mathrm{d}x} = 2a(ax + b - y)$$

We can now make *imperceptible* changes to the data points to make the loss grow.

Attacks

Random noise Does not work

FGSM Good but can be well defended by training the network with adversarial samples

Iterative FGSM Higher error than FGSM for an equivalent ε but less transferability. I-FGSM produces weaker black-box attacks.

Targeted FGSM Aims at fooling a model into outputting a given target class.

RAND + FGSM Significant improvements against adversarially trained models. RAND+FGSM transfers at lower rates than FGSM examples. Unsing RAND+FGSM to adverarially train networks does not improve their defense against RAND+FGSM.

Fast Gradient Sign Method

Move along the derivate away from the correct value as a way to maximise the error.

Black box attack

This is nice but happens if you cannot access the gradients

Adversarial examples in the physical world

This is nice but in the physical world, we are not feeding the network with our own data, it is acquired by the network's system (using camera for example).

Defense