Adversarial examples in deep learning

G. Châtel

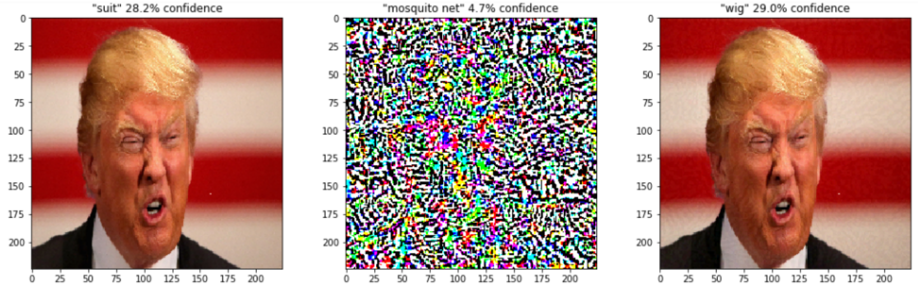
06/07/2017

What is an 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.

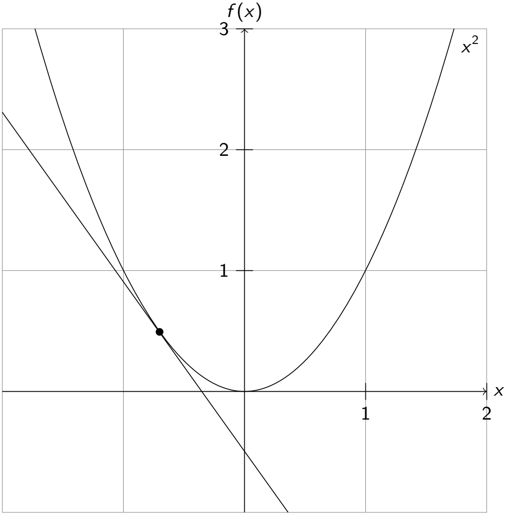
What is an adversarial example?

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

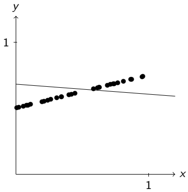
Basic concept



The curve needs to be *smooth enough* for the gradient descent to work.

Gradient descent

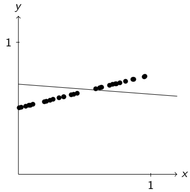
Model optimization



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.

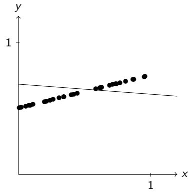
$$y = ax + b$$

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.

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

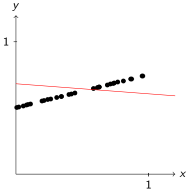
We compute how the loss is affected by small changes of a and b.

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

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

Gradient descent

Being evil

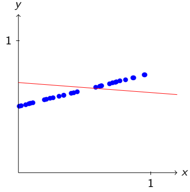


In our previous example, we have modified the model in order to minimize the loss.

$$y = ax + b$$

Gradient descent

Being evil



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

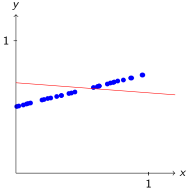
$$y = ax + b$$

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 an input that will increase the loss value.

Neural networks

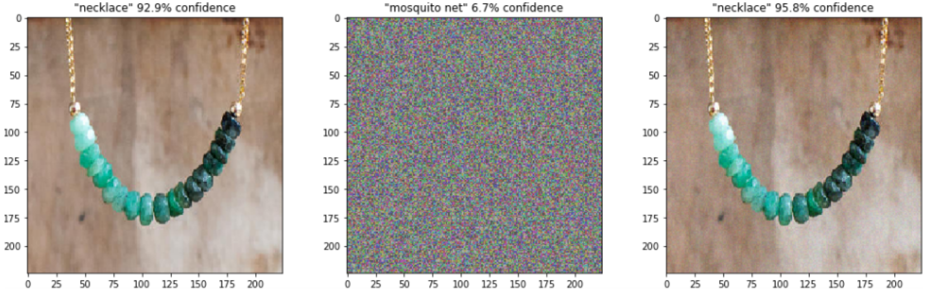
Everything works the same way when working with a neural network on an image classification task.

We also have a differentiable loss function (often categorical cross entropy) and inputs (pixel values in the case of images) that we can modify to increase the loss.

Random noise perturbation

Random noise perturbation

Nope.



Fast Gradient Sign Method [Goodfellow et al. 2015]

Let x be the original image, θ the parameters of the model, y the target associated with x and $J(\theta, x, y)$ the loss function.

We compute the gradient of the loss function according to the input pixels.

$$\nabla_x J(\theta, x, y)$$

The perturbation is the signs of these derivatives multiplied by a small number ε.

$$\eta = \varepsilon \text{sign}(\nabla_x J(\theta, x, y))$$

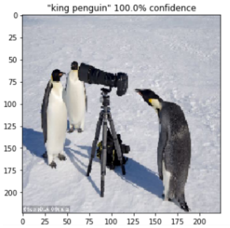
The final adversarial sample is the sum of the original image and the pertubation.

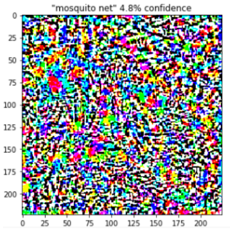
$$x_{adv} = x + \eta$$

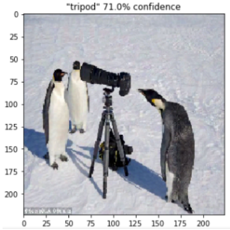
Fast Gradient Sign Method

Using the **VGG16** network with imagenet weights

$$x \quad + \quad \varepsilon \mathrm{sign}(\nabla_x J(\theta, x, y)) \quad = \quad x_{\mathrm{adv}}$$







Black box attack [Papernot et al. 2016]

or good luck getting gradients out of your self-driving car

Black box attack [Papernot et al. 2016]

Transferability of adversarial samples

We can train a new model M' to solve the same classification task as the target model M.

Black box attack [Papernot et al. 2016]

Transferability of adversarial samples

We can train a new model M' to solve the same classification task as the target model M.

Once trained, we can create an adversarial sample x'_{adv} for the M' model and experiences have shown that x'_{adv} will also fool M very often.

Black box attack [Papernot et al. 2016]

Transferability of adversarial samples

We can train a new model M' to solve the same classification task as the target model M.

Once trained, we can create an adversarial sample x'_{adv} for the M' model and experiences have shown that x'_{adv} will also fool M very often.

What if we do not have a training set for the target network? Well... build one using M predictions.

Black box attack [Papernot et al. 2016]

Transferability of adversarial samples

We can train a new model M' to solve the same classification task as the target model M.

Once trained, we can create an adversarial sample x'_{adv} for the M' model and experiences have shown that x'_{adv} will also fool M very often.

What if we do not have a training set for the target network? Well... build one using M predictions.

"After labeling 6,400 synthetic inputs to train our substitute (an order of magnitude smaller than the training set used by MetaMind) we find that their DNN misclassifies adversarial examples crafted with our substitute at a rate of 84.24%"

- Papernot et al., about their attack on the MetaMind deep neural network.

Adversarial examples in the physical world [Kurakin et al. 2017]

Being evil, for real

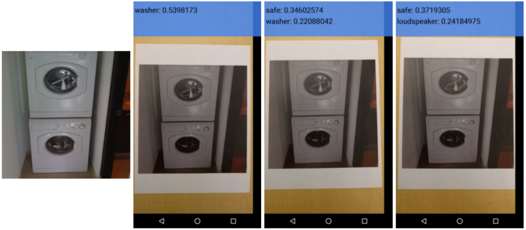
In real world scenarios, we do not run the network on our own data, it is acquired by the network's system (*e.g.* a camera).

Adversarial examples in the physical world [Kurakin et al. 2017]

Being evil, for real

In real world scenarios, we do not run the network on our own data, it is acquired by the network's system (*e.g.* a camera).

It also works, for free.



(a) Image from dataset (b) Clean image (c) Adv. image, $\epsilon = 4$ (d) Adv. image, $\epsilon = 8$

"We used images taken from a cell-phone camera as a input to an Inception v3 image classification neural network. We showed that in such a set-up, a significant fraction of adversarial images crafted using the original network are misclassified even when fed to the classifier through the camera."

Kurakin et al.

Defenses

Adversarial sample detection We try to detect whether an input sample is adversarial or not before classifying it.

Regularization Training with an adversarial objective function is an effective regularizer (from [explaining and harnessing]).

Gradient masking The goal of gradient masking is to leave the decision boundaries untouched but damage the gradient used in white-box attacks.

Distillation and network saturation These methods are used to introduce numerical instabilities in gradient computations.

Defending machine learning

"Most defenses against adversarial examples that have been proposed so far just do not work very well at all, but the ones that do work are not adaptive. This means it is like they are playing a game of whack-a-mole: they close some vulnerabilities, but leave others open."

- Ian Goodfellow, Nicolas Papernot, February 2017

References

1. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.
2. Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In Security and Privacy (SP), 2016 IEEE Symposium on (pp. 582-597). IEEE.
3. Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533.
4. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2016). Practical black-box attacks against deep learning systems using adversarial examples. arXiv preprint arXiv:1602.02697.
5. Tramèr, F., Kurakin, A., Papernot, N., Boneh, D., & McDaniel, P. (2017). Ensemble Adversarial Training: Attacks and Defenses. arXiv preprint arXiv:1705.07204.