Adversarial Attacks

Alžbeta Gogoláková
Papers

- Explaining and Harnessing Adversarial Examples
- Intriguing properties of neural networks
- Adversarial Patch

- AT-GAN: An Adversarial Generator Model for Non-constrained Adversarial Examples
What are adversarial examples?
What are adversarial examples?

They are **special inputs** that an attacker creates with a single purpose and that is to **fool** selected **machine learning model** so it will misclassify them.
"some models are vulnerable to adversarial examples"

Szegedy et al. Intriguing properties of neural networks

"linear behaviour in high-dimensional subspaces is sufficient"

Goodfellow et al. Explaining and harnessing adversarial examples

"it’s due to extreme non-linearity" (not true)
The linear explanation of adversarial examples
Precision of features is limited

36.51°C  →  36.5°C  →  36.5°C

36.52°C  →  36.5°C  →  36.5°C

precision = 0.1°C
Precision of features is limited
Classifier should respond the same

\[ x' = x + \eta \]

precision of the features = 0.1

\[ \eta = \begin{pmatrix} 0.01 & 0.059 & -0.03 \\ 0.09 & -0.07 & 0.01 \\ -0.07 & 0.08 & 0.012 \\ 0.099 & 0.05 & 0.014 \end{pmatrix} \]
Because the precision of the features is limited, it is not rational for the classifier to respond differently to an input $\mathbf{x}$ than to an adversarial input $\mathbf{x}' = \mathbf{x} + \eta$ if every element of the perturbation $\eta$ is in absolute value smaller than the precision of the features.
Max Norm

let x be a vector such that
\[ x = (x_1, x_2, \ldots, x_n) \]
then
\[ \|x\|_{\infty} := \max(|x_1|, |x_2|, \ldots, |x_n|) \]
\[ \|(1, -7, 3, -2)\|_{\infty} = 7 \]
Formalization of the condition

Condition

“every element of $\eta$ is smaller than the precision of the features” can be formally written as

$$\|\eta\|_\infty \leq \varepsilon$$
Change in the activation function

input: $x$
activation: $w^T x$

input: $x' = x + \eta$
activation: $w^T x' = w^T (x + \eta) = w^T x + w^T \eta$

We want to maximize this increase!
How to maximize increase

constraint: \( \|\eta\|_\infty \leq \varepsilon \)

we get maximal possible value from \( w^T\eta \) with respect to the max norm constraint on \( \eta \) if we assign

\[ \eta = \varepsilon \text{ sign}(w) \]
Quick example

\[ w = (30, -7, 6, -1) \]

\[ \text{sign}(w) = (1, -1, 1, -1) \]

\[ \eta = \varepsilon \text{ sign}(w) = (\varepsilon, -\varepsilon, \varepsilon, -\varepsilon) \]

\[ w^\top \eta = (30, -7, 6, -1) \begin{pmatrix} \varepsilon \\ -\varepsilon \\ \varepsilon \\ -\varepsilon \end{pmatrix} = 30\varepsilon + 7\varepsilon + 6\varepsilon + \varepsilon \]

\[ \|\varepsilon \text{ sign}(w)\|_\infty = \varepsilon \leq \varepsilon \]
Small adjustment

Maximized increase in the activation function can be expressed as:

\[ w^T \varepsilon \text{sign}(w) = \varepsilon^{mn} \]

- Average magnitude of an element of the weight vector
- Input dimensionality
THE BIG REVELATION!
1. $\|n\|_\infty$ does not grow with the dimensionality of the problem

2. $w^T \eta = \varepsilon m n$ can grow linearly with the dimensionality of the problem

For high dimensional problems, we can make many very small changes to the input that add up to one large change to the activation function.
Conclusion

A simple linear model can have adversarial examples if its input has sufficient dimensionality.
Fast Gradient Sign Method

\[ \eta = \varepsilon \sign(\nabla_x J(\theta, x, y)) \]

- \( x \) - original input image
- \( y \) - original input label
- \( \theta \) - model parameters
- \( J \) - loss function
Fast Gradient Sign Method

\[ x' = x + \varepsilon \text{ sign}(\nabla_x J(\theta, x, y)) \]
Does it look familiar?

Fast Gradient Sign Method

\[ x' = x + \epsilon \text{ sign}(\nabla_x J(\theta, x, y)) \]

Gradient descent

\[ w' = w - \alpha (\nabla_w \text{RSS}(w)) \]
Adversarial example using FGSM:
https://www.tensorflow.org/tutorials/generative/adversarial_fgsms

```python
loss_object = tf.keras.losses.CategoricalCrossentropy()

def create_adversarial_pattern(input_image, input_label):
    with tf.GradientTape() as tape:
        tape.watch(input_image)
        prediction = pretrained_model(input_image)
        loss = loss_object(input_label, prediction)

        # Get the gradients of the loss w.r.t to the input image.
        gradient = tape.gradient(loss, input_image)
        # Get the sign of the gradients to create the perturbation
        signed_grad = tf.sign(gradient)
        return signed_grad
```
Timeline

2014  2015  2019

Stop  No cycling

“new approach to generating adversarial examples”

Xiaosen Wang et al.
AT-GAN: An Adversarial Generator Model for Non-constrained Adversarial Examples
Limits of perturbation based examples

- generated by Fast Gradient Sign Method
- created from existing examples by adding a noise to it
- the bigger the $\varepsilon$ is, the higher is the success rate
- if $\varepsilon$ is large, the resulting images appear obviously altered

\[ \varepsilon = 0.01 \quad \varepsilon = 0.07 \quad \varepsilon = 0.1 \]
AT-GAN

- uses GANs to generate completely new images that are also adversarial examples
- training consists of two parts
QUESTION TIME