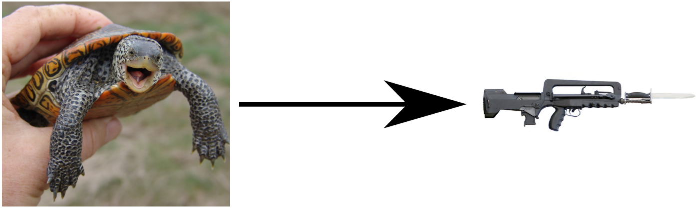
A security study of Neural ODEs

Julien Girard, Guillaume Charpiat, Zakaria Chihani, Marc Schoenauer

4 juin 2019

Outline

What are adversarial examples ?



What are adversarial examples ?

A small video to begin with

Formal definition

For an input $\mathbf{x}$, a classification function f, an adversarial perturbation δ :

maximize classifier misclassification

such that perturbation stays below a certain threshold

Formal definition

For an input **x**, a classification function f, an adversarial perturbation δ :

maximize $f(x) \neq f(x + \delta)$

such that $\|\delta\|_p \leq \varepsilon$

Why are adversarial examples important ?

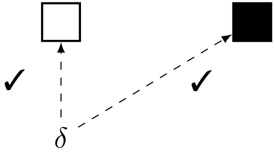
Adversarial examples :

- are transferable (Papernot et al., 2016, Transferability in Machine Learning. . ., Carlini et al. papers) $\Rightarrow$ make ML more robust

Why are adversarial examples important ?

Adversarial examples :

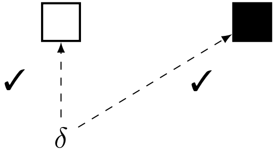
- are transferable (Papernot et al., 2016, Transferability in Machine Learning. . ., Carlini et al. papers) ⇒ make ML more robust



Why are adversarial examples important ?

Adversarial examples :

- are transferable (Papernot et al., 2016, Transferability in Machine Learning. . ., Carlini et al. papers) ⇒ make ML more robust

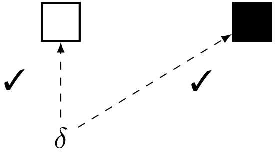


- not well understood (Goodfellow et al. 2018, Adversarial Spheres, Madry et al., 2018, Adversarial Examples are not bugs. . .) ⇒ design better ML algorithms

Why are adversarial examples important ?

Adversarial examples :

- are transferable (Papernot et al., 2016, Transferability in Machine Learning. . ., Carlini et al. papers) ⇒ make ML more robust



- not well understood (Goodfellow et al. 2018, Adversarial Spheres, Madry et al., 2018, Adversarial Examples are not bugs. . .) ⇒ design better ML algorithms
- provide us a *specification* to verify against ⇒ formal methods (later on my thesis, tomorrow at ForMaL)

Why do we bother about new architectures ?

- evaluation of new architecture designs robustness $\Rightarrow$ test state of the art attacks

Why do we bother about new architectures ?

- evaluation of new architecture designs robustness $\Rightarrow$ test state of the art attacks

Why do we bother about new architectures ?

- evaluation of new architecture designs robustness $\Rightarrow$ test state of the art attacks
- new vision on neural network computation $\Rightarrow$ better intrinsic robustness properties ?

Why do we bother about new architectures ?

- evaluation of new architecture designs robustness $\Rightarrow$ test state of the art attacks
- new vision on neural network computation $\Rightarrow$ better intrinsic robustness properties ?
- new design could inspire us to invent new attacks and defenses $\Rightarrow$ invariants as stability ?

What are ODE Nets ?

What are ODEs ?

Small recap on ODEs

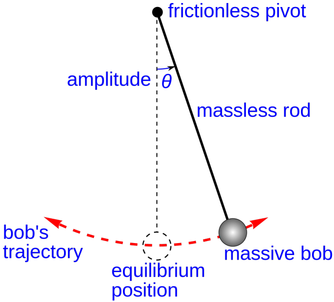
Let $\mathbf{y} : \mathbf{x} \in \mathbb{R}^d \to \mathbf{y}(\mathbf{x}) \in \mathbb{R}^p$, differentiable, t time
An ordinary differential equation (ODE) is $\mathbf{F}$ such that :

$$\mathbf{F}(\mathbf{x}, \mathbf{y}, \mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^{(n)}, t) = 0$$

Small recap on ODEs

Let $\mathbf{y} : \mathbf{x} \in \mathbb{R}^d \to \mathbf{y}(\mathbf{x}) \in \mathbb{R}^p$, differentiable, t time
An ordinary differential equation (ODE) is $\mathbf{F}$ such that :

$$\mathbf{F}(\mathbf{x}, \mathbf{y}, \mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^{(n)}, t) = 0$$



$$\begin{bmatrix} \dot{\theta} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ -\frac{g}{L}\sin(\theta(t)) \end{bmatrix}$$

$$\ddot{\theta} - \frac{g}{L}\sin(\theta(t)) = 0$$

$$\mathbf{F}(\theta, \ddot{\theta}, t) : \ddot{\theta} - \tfrac{g}{L}\sin(\theta(t))$$

How do we solve them

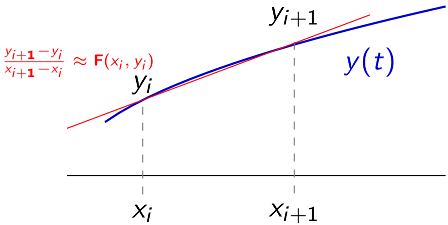
$$\ddot{y} - \epsilon * w * (1 - y^2) * \dot{y} + w^2 * y = 0$$

Van der Pol oscillator

No analytical solution in the general case $\Rightarrow$ numeric approximations

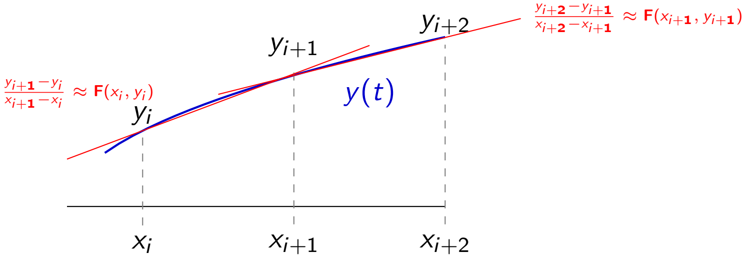
How do we solve them - continued

A simple numerical method : Euler method



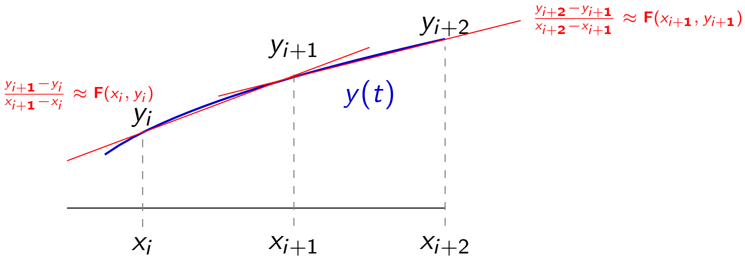
How do we solve them - continued

A simple numerical method : Euler method



How do we solve them - continued

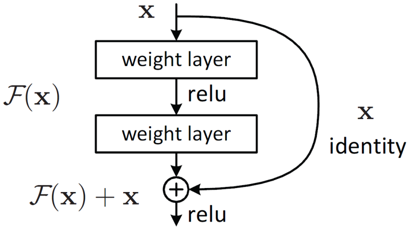
A simple numerical method : Euler method



Parameters :

- timesteps : accuracy vs speed
- for other solvers : multiple evaluations for increased stability
- error control

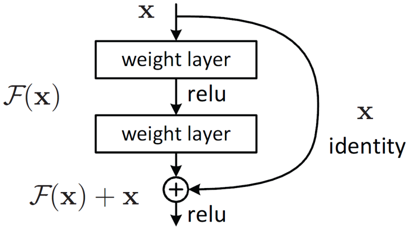
ODEs in neural networks ?



A skip connection (He et al., 2015, Residual Deep Learning. . .)

$$\mathbf{h}(\mathbf{x_i}) = \mathcal{F}(\mathbf{x_i}) + \mathbf{x_i}$$: state at a given point i

ODEs in neural networks ?

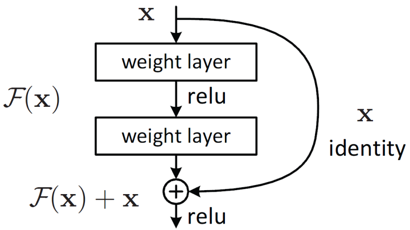


A skip connection (He et al., 2015, Residual Deep Learning. . .)

$\mathbf{h}(\mathbf{x_i}) = \mathcal{F}(\mathbf{x_i}) + \mathbf{x_i}$: state at a given point i

$\mathbf{x_i}$ is the result of previous computations $\mathbf{h}(\mathbf{x_{i-1}})$

ODEs in neural networks ?

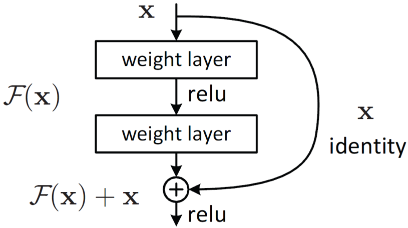


A skip connection (He et al., 2015, Residual Deep Learning. . .)

$\mathbf{h}(\mathbf{x_i}) = \mathcal{F}(\mathbf{x_i}) + \mathbf{h}(\mathbf{x_{i-1}})$: state at a given point i
$\mathbf{x_i}$ is the result of previous computations $\mathbf{h}(\mathbf{x_{i-1}})$

$$\frac{h(x_i)-h(x_{i-1})}{i+1-i} = \mathcal{F}(x_i)$$

ODEs in neural networks ?



A skip connection (He et al., 2015, Residual Deep Learning. . .)

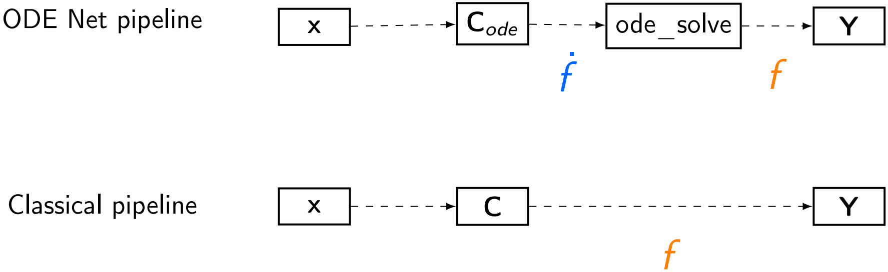
$\mathbf{h(x_i)} = \mathcal{F}(\mathbf{x_i}) + \mathbf{h(x_{i-1})}$: state at a given point i

$\mathbf{x_i}$ is the result of previous computations $\mathbf{h(x_{i-1})}$

$$\frac{h(x_i) - h(x_{i-1})}{i+1-i} = \mathcal{F}(x_i) \qquad \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \approx \mathbf{F}(x_i, y_i)$$

Euler method !

Neural Ordinary Differential Equations



Interest in image classification : lower parameter footprint

Chen et al., 2018, Neural Ordinary Differential Equations

Threat model

Goal : assert the attack and defense perimeter

Threat model

Goal : assert the attack and defense perimeter

	White box	**Black box**
Access to the model's parameters	✓	✗
Access to the model's output	✓	limited
Access to the gradient	✓	✗
Knowledge of the defense	✓	✓
Perturbation characteristics	✓	✓

Assumptions on attacker's capabilities

The Big Questions

What makes a good attack ?

What makes a good defense ?

The Big Questions

What makes a good attack ?

Break robustness within the given threat model

What makes a good defense ?

The Big Questions

What makes a good attack ?

Break robustness within the given threat model

What makes a good defense ?

1. Provably increase robustness within the given threat model
2. Limits the attack surface

FGSM (Goodfellow et al., 2014)

$$\mathsf{x}' = \mathsf{x} + \varepsilon \underbrace{\mathrm{sign}(\nabla_x L(\theta, \mathsf{x}, \mathsf{y}))}_{\delta}$$

Idea : make a step towards the direction maximizing the loss

Projected Gradient Descent l_∞ (Madry et al., 2017)

$min_\theta(\rho(\theta))$, where $\rho(\theta) = E_{(x,y)\in D}\left[max_{\delta\in D}(L(\theta, x+\delta, y))\right]$

$$x_{t+1} = \Pi(x+\delta)(x_t + \alpha \underbrace{\mathrm{sign}(\nabla_x L(\theta, x, y)}_{\delta})$$

Multiples iterations, works because of the geometric landscape

Carlini-Wagner l_2 (Carlini et al., 2016)

maximize classifier misclassification

such that perturbation stays below a certain threshold

$$min(c * \|\delta\|_p + J(x + \delta, l))$$
$$w.r.t.$$
$$x + \delta \in X$$

$$J(x, l) = max((max_{i \neq t}(logits(x)_i) - logits(x)_t), 0)$$

Carlini-Wagner l_2 (Carlini et al., 2016)

maximize $f(x) \neq f(x+\delta)$

such that $\|\delta\|_p \leq \varepsilon$

$$min(c * \|\delta\|_p + J(x+\delta, l))$$
$$w.r.t.$$
$$x + \delta \in X$$

$$J(x,l) = max((max_{i \neq t}(logits(x)_i) - logits(x)_t), 0)$$

ODE Nets are more vulnerables

	FGSM ($\varepsilon = 0.3$)	C&W	PGD ($\varepsilon = 0.3$)
classical mnist	6.23/5.19%	2.39/0.21%	5.62/4.17%
ODE mnist	6.02/8.56%	0.82/0.01%	4.54/4.34%
	FGSM ($\varepsilon = 0.1$)	C&W	PGD ($\varepsilon = 0.1$)
classical cifar10	6.53/9.96%	1.07/0.0%	5.30/0.1%
ODE cifar10	7.22/0.14%	1.06/0.0%	6.48/0.03%

Visual clues

PGD (classical)

CW (classical)

FGSM (classical)

FGSM (ODE)

PGD (ODE)

CW (ODE)

Adversarial training is still efficient

	Natural	FGSM ($\varepsilon = 0.3$)	C&W	PGD ($\varepsilon = 0.3$)
LeNet5	–/98.65%	5.76/95.81%	0.57/79.43%	5.09/97.5%
ODE	–/99.4%	6.03/96.79%	2.51/22.24%	5.48/98.52%

ODE integration time : a potential key towards robustness ?

Network	Training end time	t=1	t=10	t=100	t=500
ODE-net small	10	49.45 / 0	98.64 / 9.34	26.35 / 12.83	9.94 / 8.09
ODE-net small	10-100	61.54 / 0	98.46 / 0.52	98.31 / 23.64	94.35 / 11.67
ODE-net small	100	37.58 / 0	66.43 / 0	98.52 / 13.25	72.16 / 14.16
ODE-net large	10	97.06 / 0.18	98.93 / 30.76	91.43 / 28.20	9.35 / 8.57
ODE-net large	10-100	72.84 / 0.15	99.08 / 70.67	99.11 / 85.98	94.66 / 62.29
ODE-net large	100	78.88 / 0.59	98.85 / 83.13	99.01 / 92.62	96.68 / 78.60

Courtesy of https://rajatvd.github.io/Neural-ODE-Adversarial/

Summary

1. ODE Nets are less robust than naturals comparable models
   - less parameters
   - perturbating the derivative is easier
2. Adversarial training is still efficient
3. Possible way to improve robustness
   - integration time
   - numerical stability (more robust numerical schemes, Lyapunov invariants, etc.)

Questions ?

Shoot your questions :)