Formation

Introduction au Deep Learning

Séquence 13

"Generative Adversarial Networks (GAN)... Please draw me a sheep ! "

FIDLE

cnrs UGA Université Grenoble Alpes MIAI Grenoble Alpes EFELIA

GENCI ANITI DEVLOG Resinfo sari GRICAD SIMaP cnrs IDRIS

v2.15

FIDLE

Formation

Introduction au Deep Learning

https://fidle.cnrs.fr

Powered by CNRS CRIC,
and UGA DGDSI of Grenoble, thanks !

- Course materials (pdf)
- Practical work environment
- Corrected notebooks
- Videos (YouTube)

docker
v3.0.9

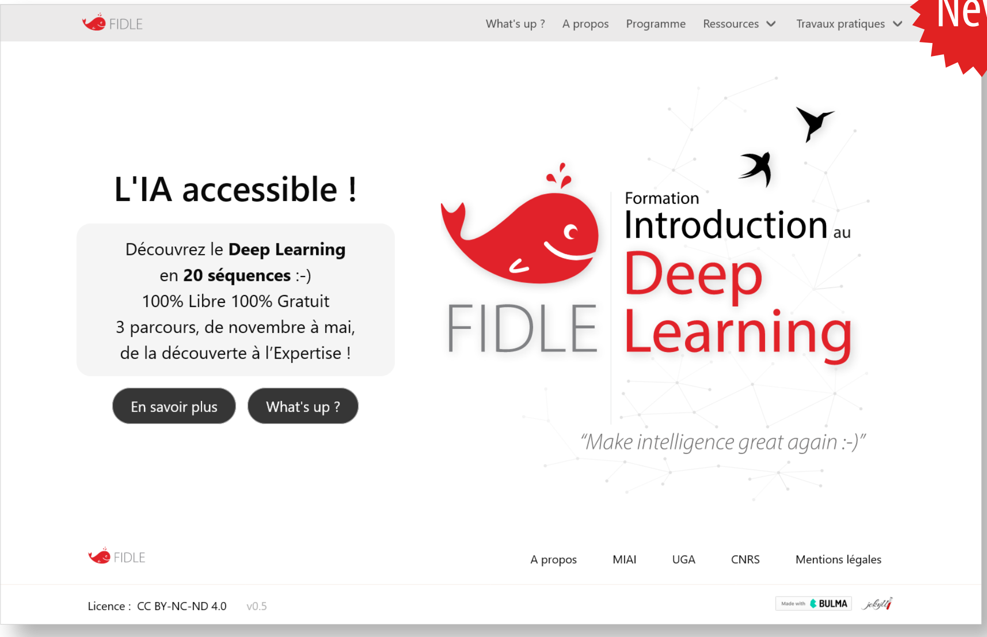
(*) Procedure via Docket or pip
Remember to get the latest version !

FIDLE

Formation

Introduction au Deep Learning

New !



https://fidle.cnrs.fr

Powered by CNRS CRIC,
and UGA DGDSI of Grenoble, thanks !

Formation
Introduction au Deep Learning

FIDLE

Questions and answers :

https://fidle.cnrs.fr/q2a

Accompanied by :
IA Support (dream) Team of IDRIS

Directed by :
Agathe, Baptiste et Yanis - UGA/DAPI
Thibaut, Kamel - IDRIS

FIDLE

Formation

Introduction au Deep Learning

FIDLE

https://fidle.cnrs.fr/listeinfo

Fidle information list

Agoria

http://fidle.cnrs.fr/agoria

AI exchange list

agoria@grenoble.cnrs.fr

(*) ESR is Enseignement Supérieur et Recherche, french universities and public academic research organizations

Formation

Introduction au Deep Learning

FIDLE

https://listes.services.cnrs.fr/wws/info/devlog

List of ESR* « Software developers » group

https://listes.math.cnrs.fr/wws/info/calcul

List of ESR* « Calcul » group

(*) ESR is Enseignement Supérieur et Recherche, french universities and public academic research organizations

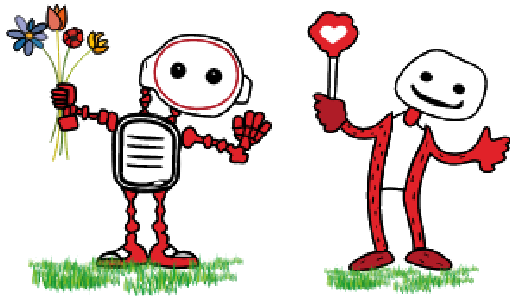
Save the dates !

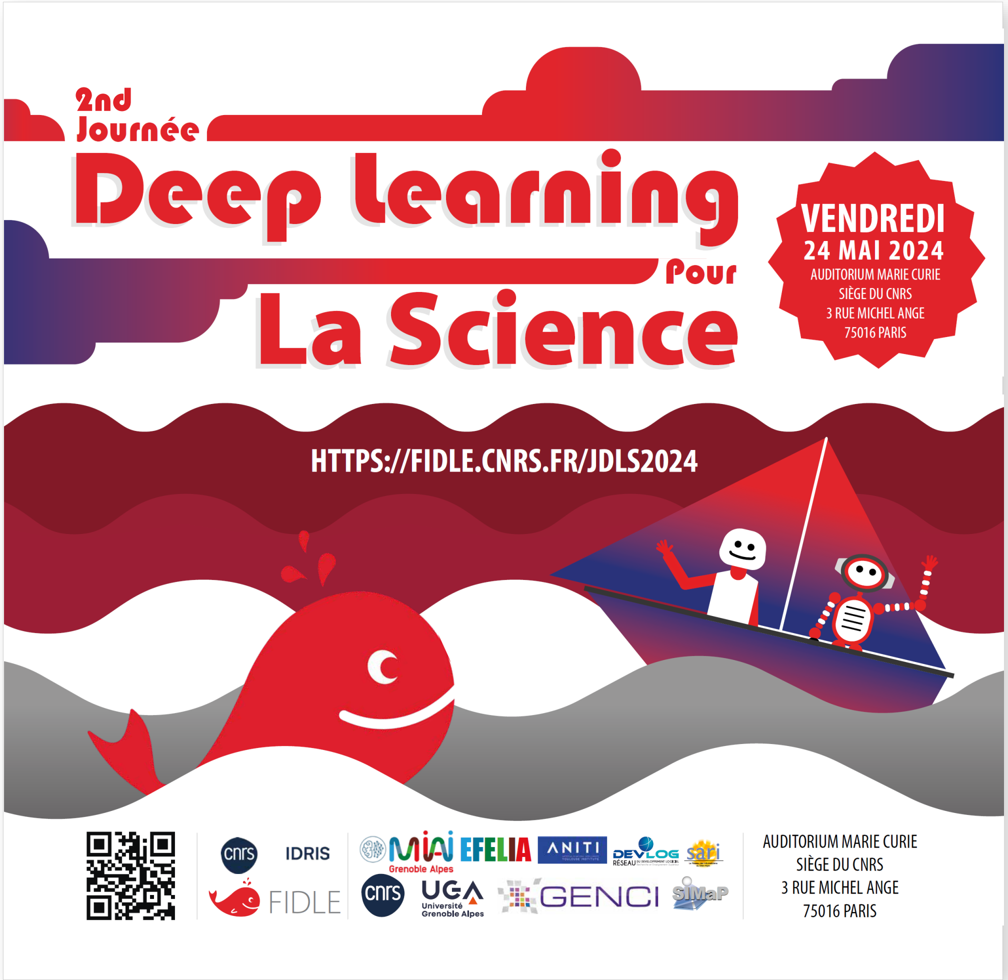
Panoram'IA

Magazine de l'actualité IA

Chaine Youtube de l'IDRIS

→ Vendredi 5 avril





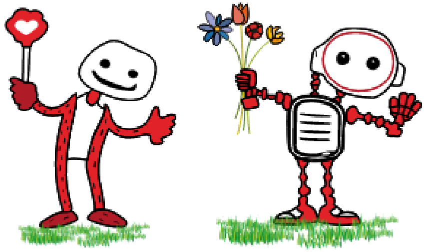
→ Vendredi 24 mai, Paris Michel Ange

Vendredi 24 mai 2024

CNRS, Paris Michel Ange

Présentions « Flash » et/ou posters ?

jdls@idris.fr



FIDLE

SAISON 23/24

Bases, Concepts et Enjeux

1 — History and Fundamental Concepts

2 — New ! — Data, models and representation's hell — Data and models

3 — New ! — Demonstration Illustration — LLM / Text to Image

4 — AI, Law, Society and Ethics

L'IA comme un outil,

5 — New ! — Mathematics, gradients everywhere !

6 — New ! — Learning methodology

7 — Convolutional models — CNN

8 — Sparse (text) and sequences data — Embedding, RNN

9 — «Attention is All You Need» — Transformers

10 — Graph Neural Network — GNN

11 — Autoencoder networks — AE

12 — Variational Antoencoder — VAE

13 — Generative Adversarial Networks — GAN

14 — Diffusion Model — Text to image

15 — Deep Reinforcement Learning — RL

16 — Physics-Informed Neural Networks — PINNS

Acteur de l'IA

17 — Learning faster and cheaper, Eco-Friendly

18 — Jean-Zay — GPU acceleration

19 — New ! — New models — VLM, SM, Multimodal, ...

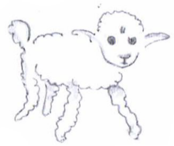
20 — New ! — Case Study — Experience feedback

13

Generative Adversarial Networks

GAN

1 About GAN
- → Principles
- → Training

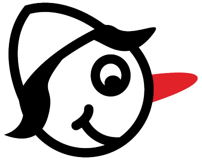


2 Example 1
- → Draw me a sheep !

3 From GANs to *GANs
- → Wasserstein and Earth Moving Distance
- → From GAN to WGAN
- → From WGAN to WGAN-GP

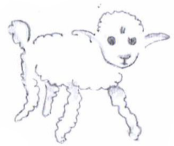
4 Example 2
- → Again, but with a WGAN-GP :-)

13



Generative Adversarial Networks

GAN



1. **About GAN**
   - → Principles
   - → Training

2. **Example 1**
   - → Draw me a sheep !

3. **From GANs to *GANs**
   - → Wasserstein and Earth Moving Distance
   - → From GAN to WGAN
   - → From WGAN to WGAN-GP

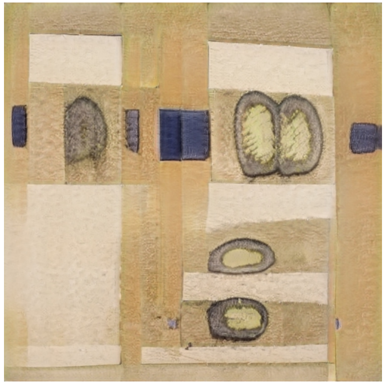
4. **Example 2**
   - → Again, but with a WGAN-GP :-)

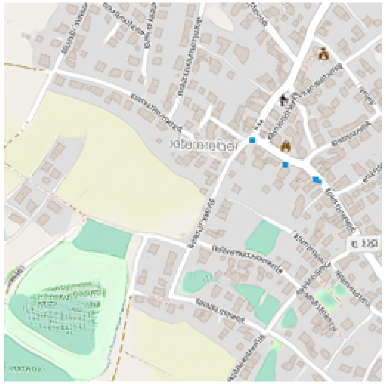
Illustration : https://www.saintexupery-domainepublic.be/

About GAN










https://thisxdoesnotexist.com/

Which Face Is Real ?





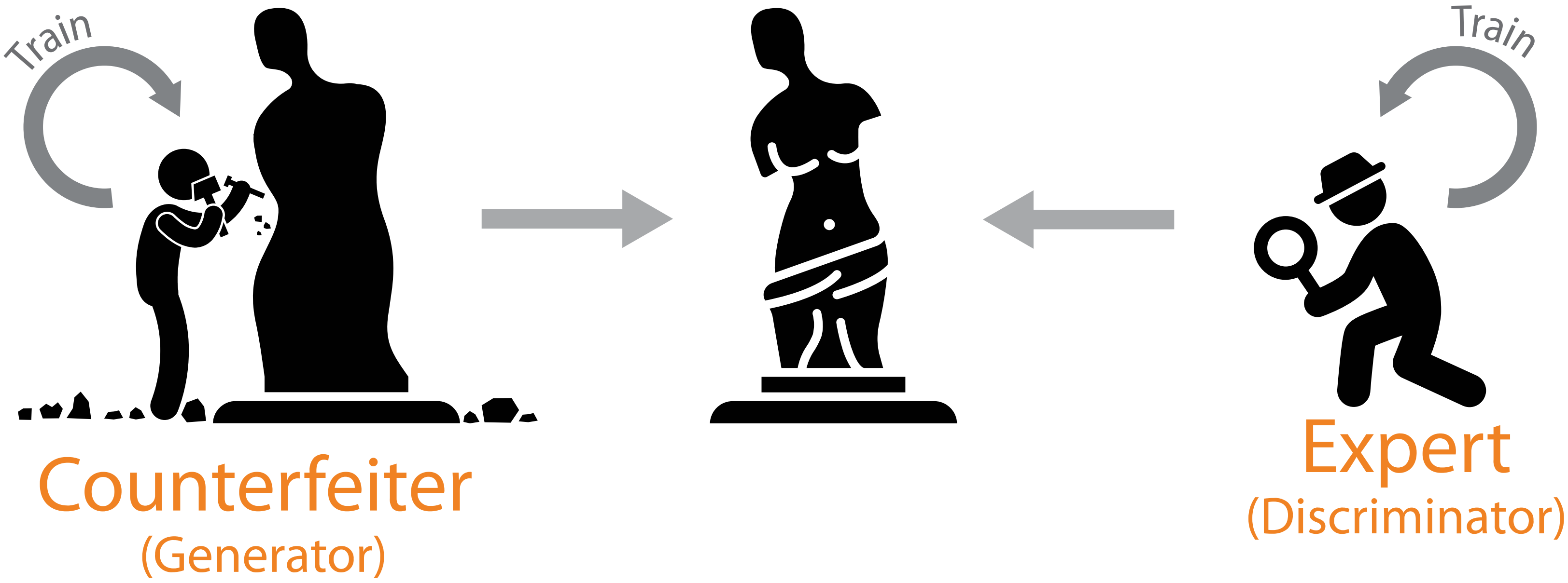



https://www.whichfaceisreal.com

Sophie J. Nightingale and Hany Farid, « AI-synthesized faces are indistinguishable from real faces and more trustworthy », PNAS, https://doi.org/10.1073/pnas.2120481119, February 2022

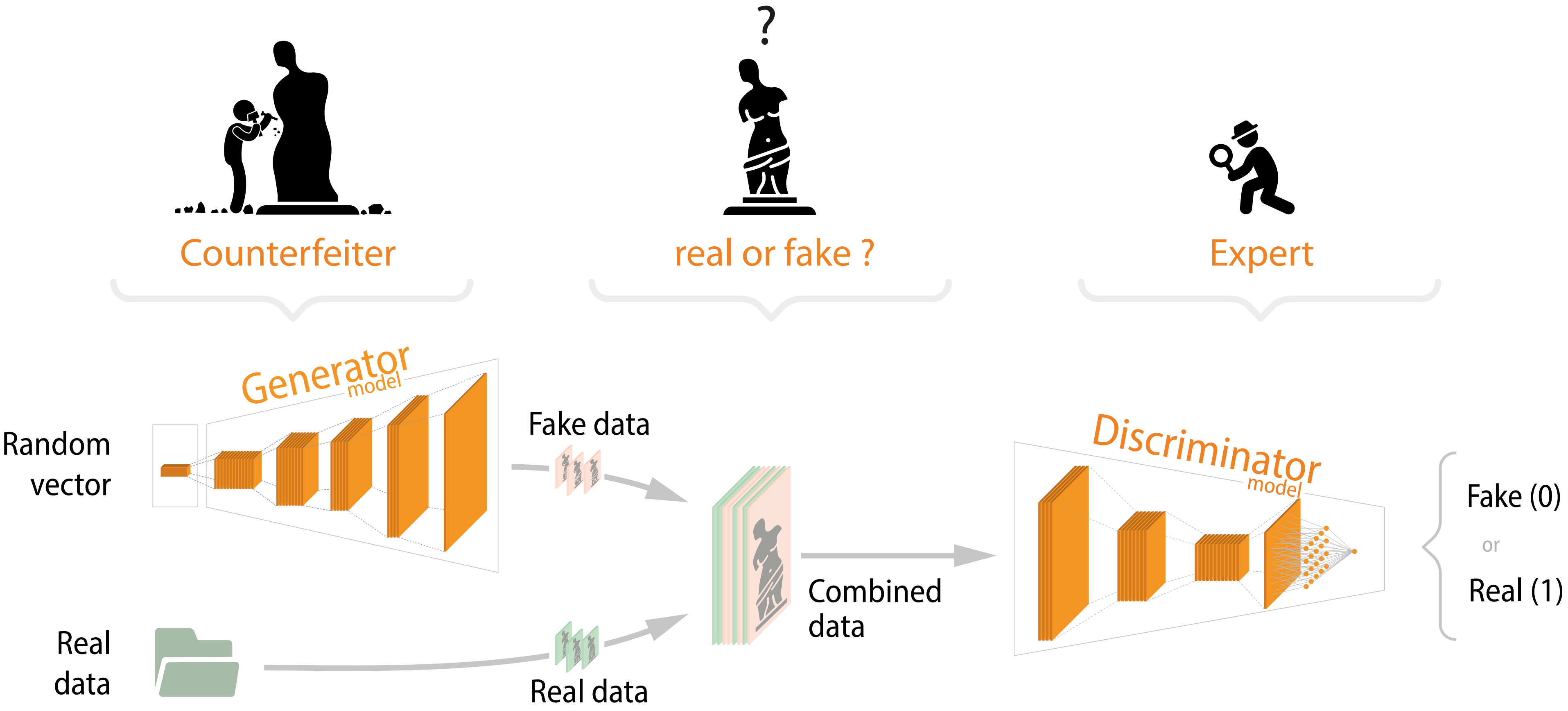
« (…) synthetically generated faces are more trustworthy than real faces. This may be because synthesized faces tend to look more like average faces which themselves are deemed more trustworthy. (…) »

About GAN



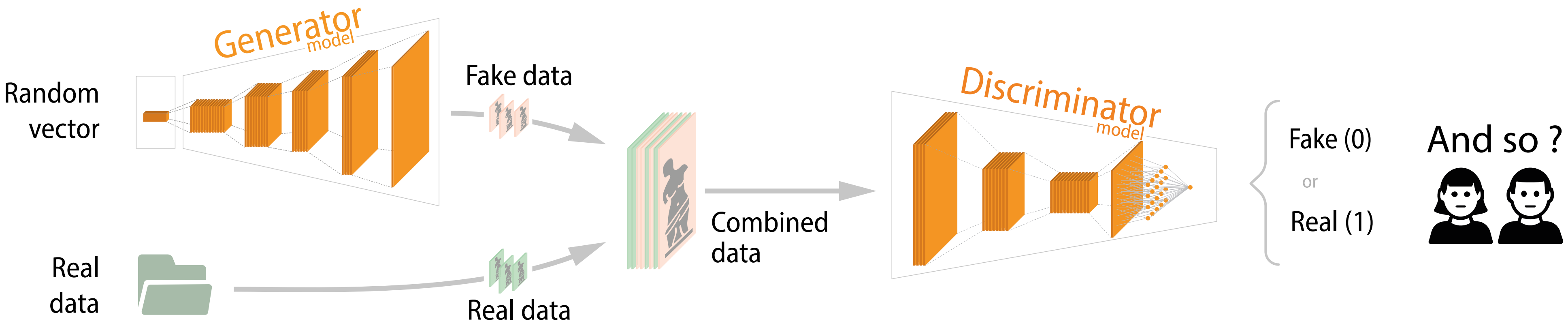
Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville et Yoshua Bengio, « Generative Adversarial Networks », in Advances in Neural Information Processing Systems 27, 2014
https://arxiv.org/abs/1406.2661

Principle



?
Counterfeiter
real or fake ?
Expert
Generator
model
Random vector
Fake data
Real data
Real data
Combined data
Discriminator
model
Fake (0)
or
Real (1)

How to train a GAN ?



A two-step learning process will be necessary :

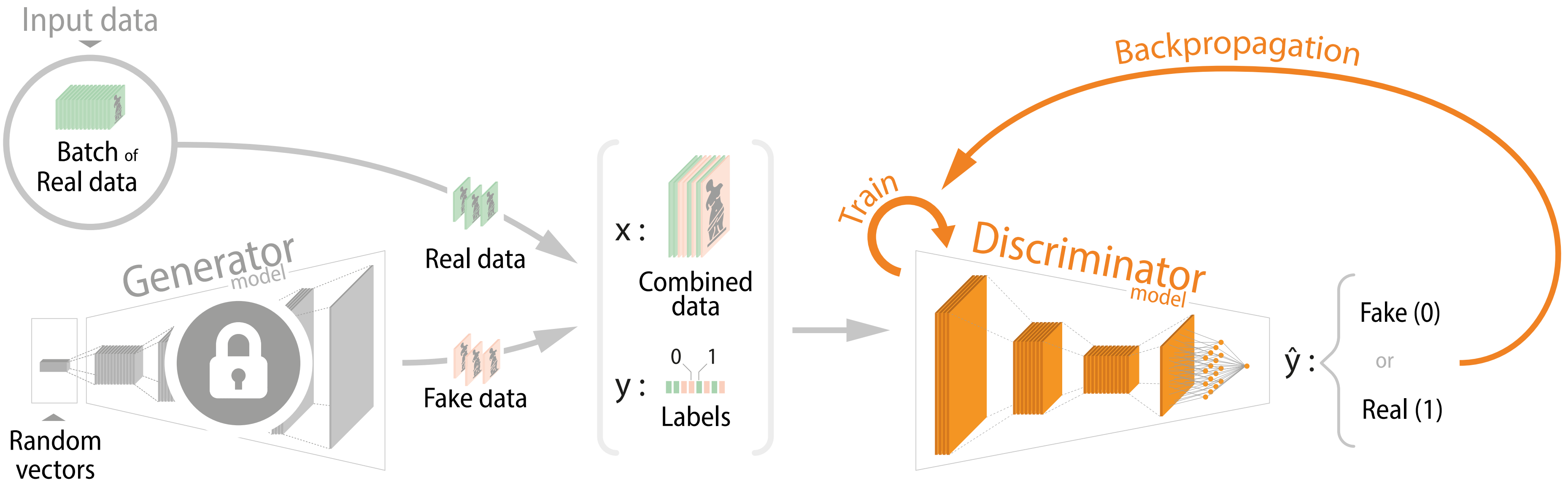
Step 1 - Update Discriminator

Step 2 - Update Generator

n epochs

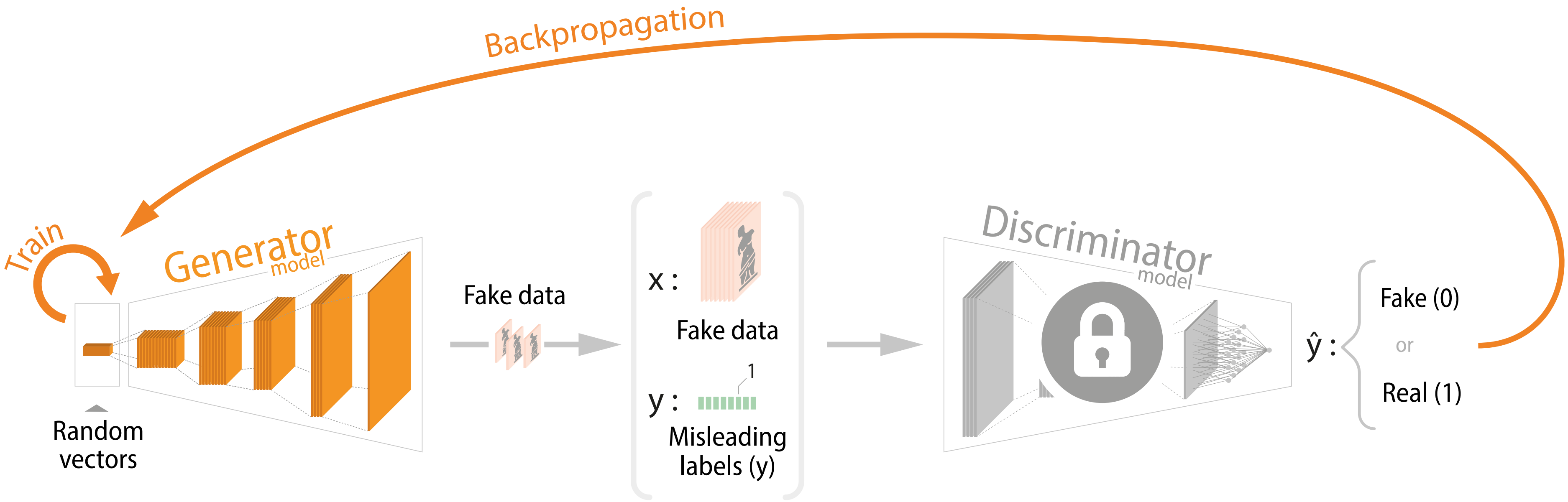
Step1 - Update Discriminator

Generator is locked



Step2 - Update Generator

Discriminator is locked



13

Generative Adversarial Networks

GAN

1 **About GAN**
- → Principles
- → Training

2 **Example 1**
- → Draw me a sheep !

3 **From GANs to *GANs**
- → Wasserstein and Earth Moving Distance
- → From GAN to WGAN
- → From WGAN to WGAN-GP

4 **Example 2**
- → Again, but with a WGAN-GP :-)

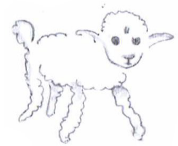
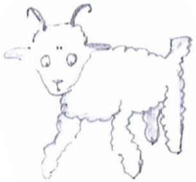


Illustration : https://www.saintexupery-domainepublic.be/

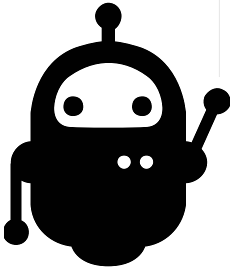


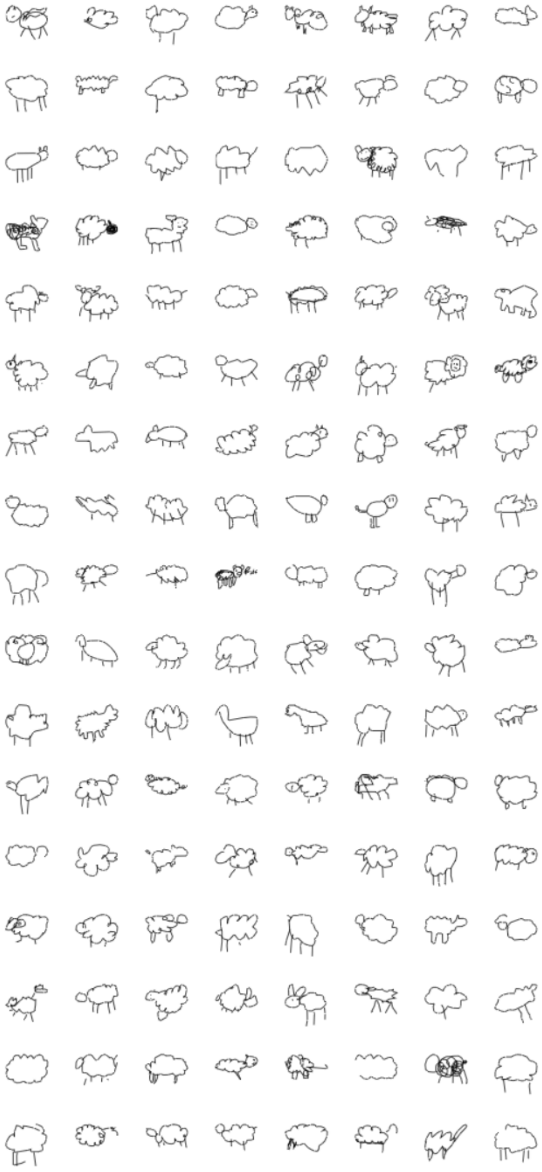
Draw me a sheep !

Notebook : [SHEEP1]

Objective :
Learn to draw or generate sheep.

Dataset :
Quick draw dataset
https://quickdraw.withgoogle.com/data





SHEEP1

START

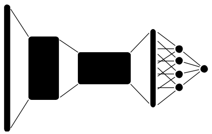
1 Import and init

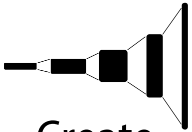
2 Parameters

```
latent_dim = 128
Scale      = 0.01
```

3 Load dataset & have a look !

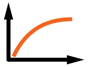
4 Create Discriminator





5 Create Generator

6 Build, compile & train



7 History

7 Drawing new sheep !

END

6m on a V100 !
(10 epochs)

Objectives :

Learn to draw sheep !

Fine, but...
How to calculate a gradient with Keras 3 and **PyTorch** ?

How to calculate its preferred gradient?

$$\begin{cases} f(x) = x^2 + 4x - 5 \\ \dot{f}(x) = \dfrac{\delta f}{\delta x} = 2x + 4 \end{cases}$$

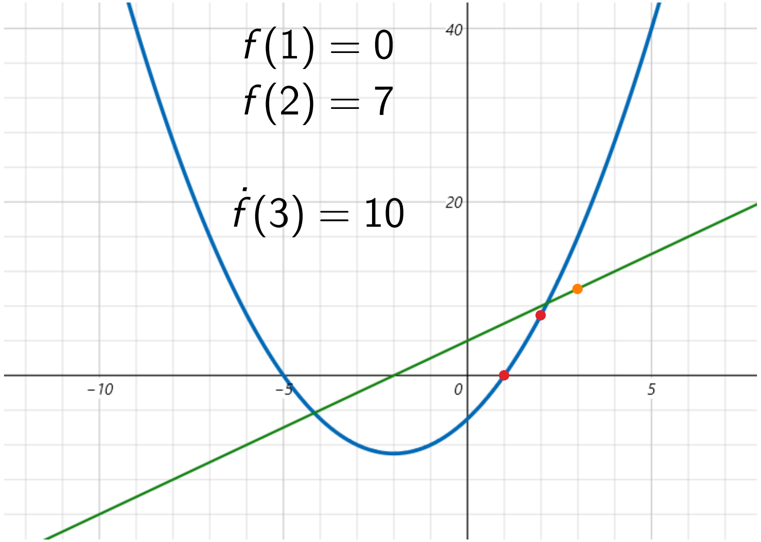
```python
import numpy as np

# ---- My function f
def f(x):
  y = x**2 + 4*x - 5
  return y

# ---- Examples :
print('f(1) is : ', f(1))
print('f(2) is : ', f(2))
```

```
f(1) is :  0
f(2) is :  7
```

We have :



How to calculate its preferred gradient?

Gradients using PyTorch

$$\begin{cases} f(x) = x^2 + 4x - 5 \\ \dot{f}(x) = \dfrac{\delta f}{\delta x} = 2x + 4 \end{cases}$$

```
Import torch

x = torch.tensor(3.0, requires_grad = True)
print("x:", x)
```

x: tensor(3., requires_grad=True)

```
y = x*x + 4*x + - 5

y.backward()

dx=x.grad
print('dx=',dx)
```

```
dx= tensor(10.)
```

We have :

$$f(1) = 0$$
$$f(2) = 7$$

$$\dot{f}(3) = 10$$

SHEEP1

6m on a V100 !
(10 epochs)

START

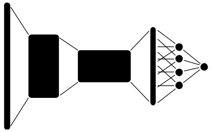
1 — Import and init

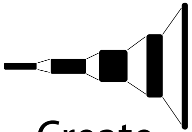
2 — Parameters

```
latent_dim = 128
Scale      = 0.01
```

3 — Load dataset & have a look !

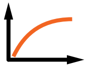
4 — Create Discriminator





5 — Create Generator

6 — Build, compile & train



7 — History

8 — Drawing new sheep !

END

Objectives :

Learn to draw sheep !

13

Generative Adversarial Networks

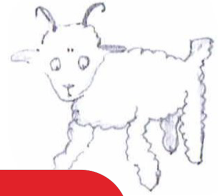
GAN

1 About GAN

- Principles
- Training

2 Example 1

- Draw me a sheep !



3 From GANs to *GANs

- Wasserstein and Earth Moving Distance
- From GAN to WGAN
- From WGAN to WGAN-GP

4 Example 2

- Again, but with a WGAN-GP :-)

Illustration : https://www.saintexupery-domainepublic.be/

Original GAN 2014

Categorical crossentropy : $H(y,\hat{y}) = -\frac{1}{n_{obs}} \sum_{i=1}^{n_{obs}} \sum_{c=1}^{n_{class}} y_c \cdot \log \hat{y}_c$

$$L_D = \frac{1}{m} \sum_{i=1}^{m} \left[\log(D(x^{(i)})) + \log(1 - D(G(z^{(i)}))) \right]$$

$$L_G = \frac{1}{m} \sum_{i=1}^{m} \left[\log(1 - D(G(z^{(i)}))) \right]$$

GAN loss functions

$x^{(i)}$	is a real data from a set of m values
$z^{(i)}$	is a latent vector from a set of m values
D(x)	Output of the discriminator for a real image x ie : probability that a real image is considered as real.
$G(z^{(i)})$	Output of the generator from an input z, from latent space so, G(z) look like an X image, but is really fake...
$D(G(z^{(i)}))$	Output of the discriminator for a fake image. ie: probability that a fake image is considered as real.

Discriminator will try to **maximize** L_D :

$$\forall x \in X, D(x) \approx 1$$

$$\forall z \in Z, D(G(z)) \approx 0$$

Generator will try to **minimize** L_G :

$$\forall z \in Z, D(G(z)) \approx 1$$

Where : $\begin{cases} \text{fake} : 0 \\ \text{real} : 1 \end{cases}$

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, « Generative Adversarial Networks », https://arxiv.org/abs/1406.2661, 2014

Original GAN 2014

Categorical crossentropy : $H(y,\hat{y}) = -\frac{1}{n_{\text{obs}}}\sum_{i=1}^{n_{\text{obs}}}\sum_{c=1}^{n_{\text{class}}} y_c \cdot \log \hat{y}_c$

$$L_D = \frac{1}{m}\sum_{i=1}^{m}\left[\log(D(x^{(i)})) + \log(1 - D(G(z^{(i)})))\right]$$
$$L_G = \frac{1}{m}\sum_{i=1}^{m}\left[\log(1 - D(G(z^{(i)})))\right]$$

GAN loss functions

Limitations :

Gradient vanishing problem.
Risk of blocking in the early stages of GAN learning, when the discriminator work is very easy.

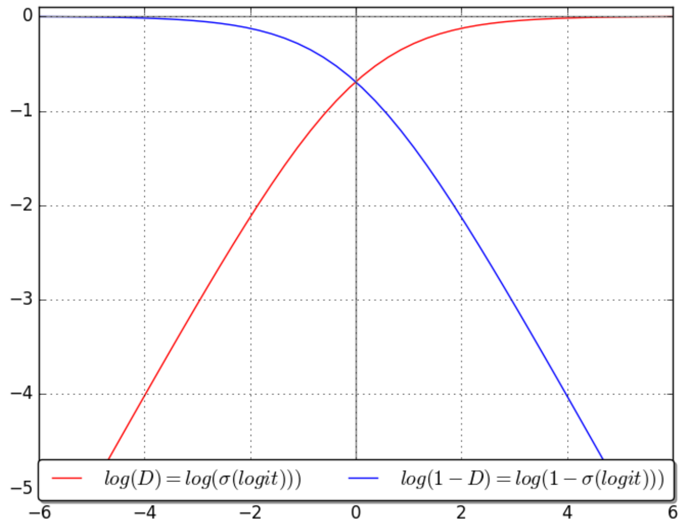


Figure 6: Illustration of vanishing gradient in the negative quadrant when using the original loss formulation log(1-D) (blue curve). The x-axis is D's logit (output of last layer before sigmoid activation). Minimising log(1-D) yields the same solution as maximising log(D) (red curve) but the red curve exhibits stronger gradients.

https://developer.nvidia.com/blog/photo-editing-generative-adversarial-networks-1/

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, « Generative Adversarial Networks », https://arxiv.org/abs/1406.2661, 2014

Wasserstein GAN 2017

→ Proposition of a more theoretical approach to the learning strategy [1]:

What is the best way to measure the distance between two distributions ?

- Total Variation distance
- Kullback-Leibler divergence
- Jensen-Shannon divergence
- Earth-Mover Distance 2] / Wasserstein-1

Optimization problem [2]

Distance du terrassier

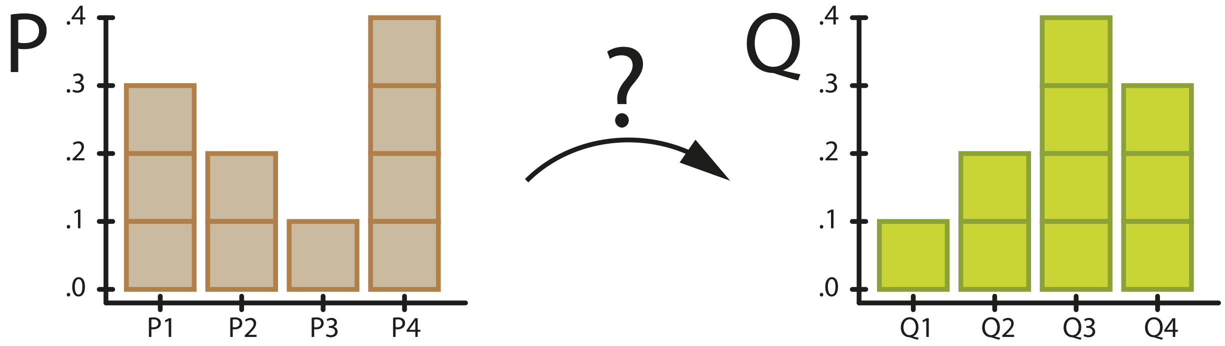
[1] Martin Arjovsky, Soumith Chintala, Léon Bottou, « Wasserstein GAN », https://arxiv.org/abs/1701.07875, 2017

[2] Cédric Villani, « Optimal transport, old and new », Springer, 2008
https://cedricvillani.org/sites/dev/files/old_images/2012/08/preprint-1.pdf
https://www.youtube.com/watch?v=zo46TEp6FB8

Wasserstein-1
Earth-Mover Distance

Objective is to measuring the distance between 2 distributions

This distance can be interpreted by the minimal energy needed to move from one box distribution to another:



This energy can be calculated as: $W = \text{Number of boxes moved} \times \text{Moving distance of the boxes}$

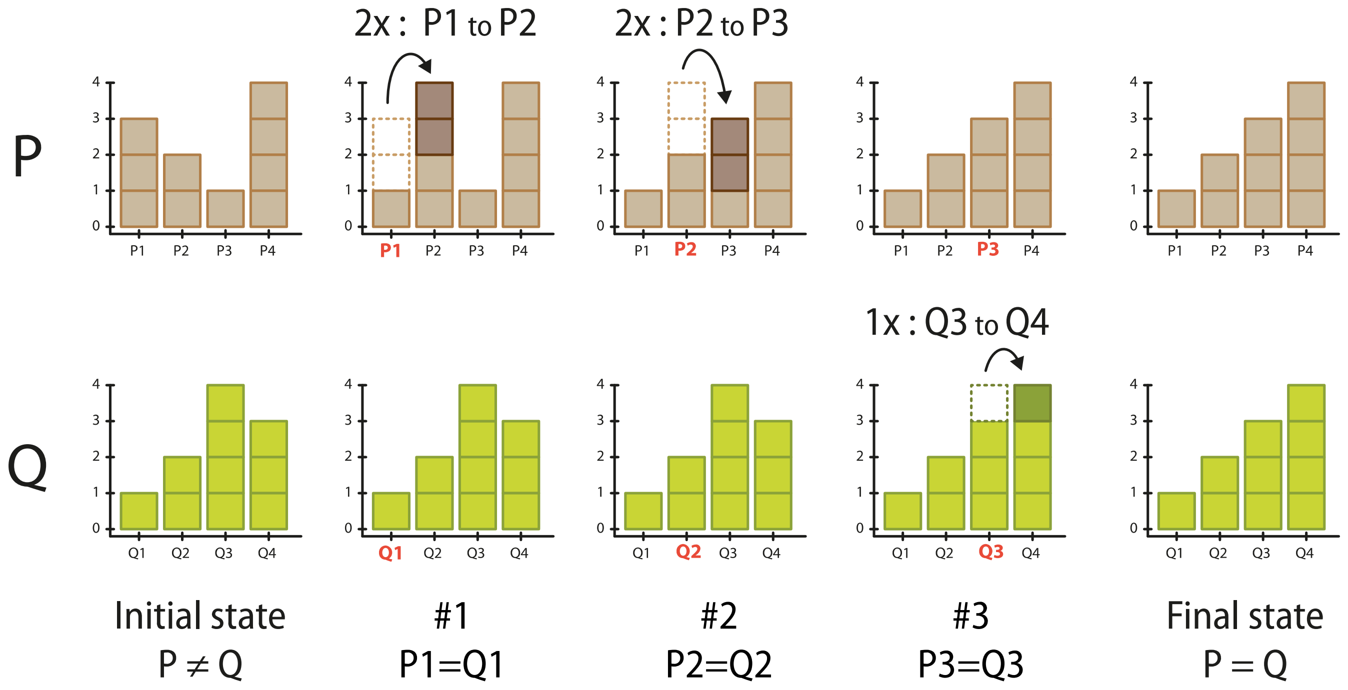
We can write :

$$W = \sum_{i}^{\text{class}} |\delta_i|$$

with $\begin{cases} \delta_0 = 0 \\ \delta_{i+1} = \delta_i + P_i - Q_i \end{cases}$

$\delta_0 = 0$
$\delta_1 = 0 + 3\text{-}1 = 2$
$\delta_2 = 2 + 2\text{-}2 = 2$
$\delta_3 = 2 + 1\text{-}4 = \text{-}1$
$\delta_4 = \text{-}1 + 4\text{-}3 = 0$

$$W = \sum_{i=1}^{4} |\delta_i| = 5$$



With a continuous probability domain,
the formula can be written :

$$\mathrm{EMD}(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, Pg)} \sum_{x,y} \gamma(x, y) \|x - y\| = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$

Earth Mover Distance

With : $\Pi(P_r, Pg)$ Set of all possible joint probability distributions between P_r and P_g

$\gamma(x, y)$ Percentage of « sand » that must be transported from point x to point y so that x follows the same probability distribution as y.

$\|x - y\|$ Distance between x and y

More on the subject:

Lilian Weng, « From GAN to WGAN », https://arxiv.org/abs/1904.08994, 2019
Ludovic Platon , « GAN : Vers une meilleure estimation des distributions » (fr), https://www.aquiladata.fr/insights/gan-vers-une-meilleure-estimation-des-distributions/

→ Wasserstein GAN :

$$L_{\text{critic}} = \frac{1}{m}\sum_{i=1}^{m}\left[D(x^{(i)}) - D(G(z^{(i)}))\right]$$

$$L_{\text{generator}} = \frac{1}{m}\sum_{i=1}^{m}\left[D(G(z^{(i)}))\right]$$

WGAN loss functions

Critic / Discriminator
try to **maximize** L_{critic}

$\forall x \in X, \forall z \in Z$
$D(x) \gg D(G(z))$

Generator
try to **maximize** $L_{generator}$

$x^{(i)}$	is a real data from a set of m values
$z^{(i)}$	is a latent vector from a set of m values
D(x)	Output of the discriminator for a real image x ie : probability that a real image is considered as real.
$G(z^{(i)})$	Output of the generator from an input z, from latent space so, G(z) look like an X image, but is really fake...
$D(G(z^{(i)}))$	Output of the discriminator for a fake image. ie: probability that a fake image is considered as real.

Note : fake : low score
real : hight score

Martin Arjovsky, Soumith Chintala, Léon Bottou, « Wasserstein GAN »,
https://arxiv.org/abs/1701.07875, 2017

→ Wasserstein GAN :

$$L_{\text{critic}} = \frac{1}{m}\sum_{i=1}^{m}\left[D(x^{(i)}) - D(G(z^{(i)}))\right]$$

$$L_{\text{generator}} = \frac{1}{m}\sum_{i=1}^{m}\left[D(G(z^{(i)}))\right]$$

WGAN
loss functions

Critic / Discriminator
try to **maximize** L_{critic}

$\forall x \in X, \forall z \in Z$
$D(x) \gg D(G(z))$

Generator
try to **maximize** $L_{generator}$

!

According to the authors, the discriminator should **no longer be considered as a classifier** but more as a **critic**. The output of the discriminator will be greater for true instances than for false instances.

Concretely, a critic has **no sigmoid** function at its output

Note : fake : low score
real : hight score

Martin Arjovsky, Soumith Chintala, Léon Bottou, « Wasserstein GAN », https://arxiv.org/abs/1701.07875, 2017

k-Lipschitz function :
$\forall (x, y) \in E^2,\ |f(x) - f(y)| \leq k\ |x - y|$

→ Wasserstein GAN :

$$L_{\text{critic}} = \frac{1}{m} \sum_{i=1}^{m} \left[D(x^{(i)}) - D(G(z^{(i)})) \right]$$

$$L_{\text{generator}} = \frac{1}{m} \sum_{i=1}^{m} \left[D(G(z^{(i)})) \right]$$

WGAN
loss functions

Critic / Discriminator
try to **maximize** L_{critic}

$\forall x \in X, \forall z \in Z$
$D(x) \gg D(G(z))$

Generator
try to **maximize** $L_{generator}$

! The calculation of the EMD distance is complex. A simplified version is however possible if the discriminator is a **k-Lipschitz function**, implying a constraint on it.

The proposed solution is to clip the weights to contain them in a **limited interval [-c,c]**, with c=0.01

Note : $\begin{cases} \text{fake : low score} \\ \text{real : hight score} \end{cases}$

Martin Arjovsky, Soumith Chintala, Léon Bottou, « Wasserstein GAN », https://arxiv.org/abs/1701.07875, 2017

→ Wasserstein GAN :

$$L_{\text{critic}} = \frac{1}{m}\sum_{i=1}^{m}\left[D(x^{(i)}) - D(G(z^{(i)}))\right]$$

$$L_{\text{generator}} = \frac{1}{m}\sum_{i=1}^{m}\left[D(G(z^{(i)}))\right]$$

WGAN
loss functions

The calculation of the EMD distance is complex. A simplified version is however possible if the discriminator is a k-Lipschitz function, implying a constraint on it.

The proposed solution is to clip the weights to contain them in a limited interval [-c,c], with c=0.01

Limitations :

« Weight clipping is a clearly terrible way to enforce a Lipschitz constraint (…) and we actively encourage interested researchers to improve on this method. » [2]

Martin Arjovsky, Soumith Chintala, Léon Bottou, « Wasserstein GAN », https://arxiv.org/abs/1701.07875, 2017

Wasserstein GAN - GP

2017 dec. 25

→ Wasserstein GAN with **Gradient Penalty** :

Here is the gradient penalty

$$L_{\text{critic}} = \underset{\tilde{x}\sim\mathbb{P}_g}{\mathbb{E}}[D(\tilde{x})] - \underset{x\sim\mathbb{P}_r}{\mathbb{E}}[D(x)] + \boxed{\lambda \underset{\hat{x}\sim\mathbb{P}_{\hat{x}}}{\mathbb{E}}[(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1)^2]}$$

$$L_{\text{generator}} = \frac{1}{m}\sum_{i=1}^{m}\left[D(G(z^{(i)}))\right]$$

WGAN-GP loss functions

x	is a real data from a set of real values
$\tilde{x} = G(z)$	is a fake data, from the generator
$\hat{x}$	mix fake / real data
λ	penalty coefficiel, proposed as λ=10
$z^{(i)}$	is a latent vector from a set of m values
D(x)	Output of the discriminator for a real image x
$G(z^{(i)})$	Output of the generator from an input z, from latent space
$D(G(z^{(i)}))$	Output of the discriminator for a fake image.

$$\hat{x} = \varepsilon x + (1-\varepsilon)\tilde{x}$$

with $t \sim U[0, 1]$

Critic / Discriminator try to minimize L_{critic}

$$\forall x \in X, \forall z \in Z$$

$$D(x) \gg D(G(z))$$

Generator try to maximize $L_{\text{generator}}$

The paper demonstrates that the addition of the penalty gradient guarantees that the loss function is 1-Lipschitz.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, « Improved Training of Wasserstein GANs », https://arxiv.org/abs/1704.00028, 2017

Wasserstein GAN - GP

Algorithm 1 WGAN with gradient penalty. We use default values of $\lambda = 10$, $n_{\text{critic}} = 5$, $\alpha = 0.0001$, $\beta_1 = 0$, $\beta_2 = 0.9$.

Require: The gradient penalty coefficient λ, the number of critic iterations per generator iteration n_{critic}, the batch size m, Adam hyperparameters α, β_1, β_2.
Require: initial critic parameters w_0, initial generator parameters θ_0.

while θ has not converged **do**
    for $t = 1, ..., n_{\text{critic}}$ **do**
        for $i = 1, ..., m$ **do**
            Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$, latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\epsilon \sim U[0, 1]$.
            $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z})$
            $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\tilde{\boldsymbol{x}}$
            $L^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
        end for
        $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2)$
    end for
    Sample a batch of latent variables $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m} \sim p(\boldsymbol{z})$.
    $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -D_w(G_\theta(\boldsymbol{z})), \theta, \alpha, \beta_1, \beta_2)$
end while

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, « Improved Training of Wasserstein GANs », https://arxiv.org/abs/1704.00028, 2017

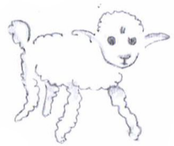
13

Generative Adversarial Networks

GAN

1 About GAN

- → Principles
- → Training



2 Example 1

- → Draw me a sheep !

3 From GANs to *GANs

- → Wasserstein and Earth Moving Distance
- → From GAN to WGAN
- → From WGAN to WGAN-GP

4 Example 2

- → Again, but with a WGAN-GP :-)

Illustration : https://www.saintexupery-domainepublic.be/

SHEEP2

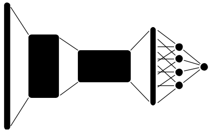
START

1. Import and init
2. Parameters

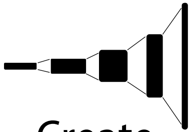
```
latent_dim  = 128
Scale       = 0.01
…
n_critic    = 2
```

3. Load dataset & have a look !

4. Create Discriminator



5. Create Generator



6. Build, compile & train

7. History

8. Drawing new sheep !

END

6m on a V100 !
(10 epochs)

Objectives :
Still learning to draw a sheep, but with a WGAN-GP !

…and what else ?

It is time to conclude ! ;-)

Illustration : https://www.saintexupery-domainepublic.be/

GANs, GANs, more GANs...

Conditional GANs
Mehdi Mirza, Simon Osindero,
« Conditional Generative Adversarial Nets »,
https://arxiv.org/abs/1411.1784, 2014

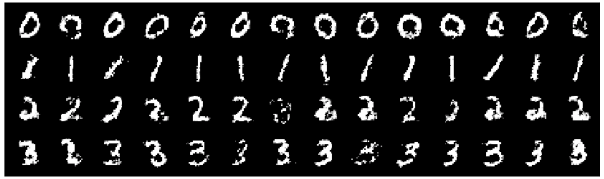
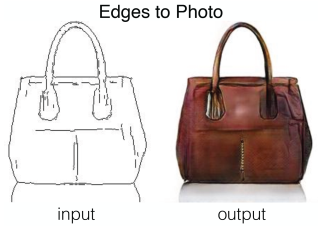
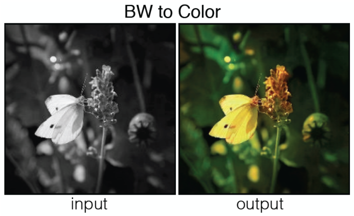


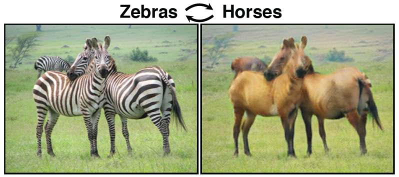
Image-to-Image Translation
Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros,
« Image-to-Image Translation with Conditional Adversarial Networks »,
https://arxiv.org/abs/1611.07004, 2016

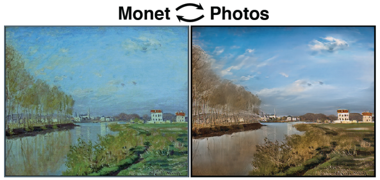




Progressive GANs
Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, « Progressive Growing of GANs for Improved Quality, Stability, and Variation »,
https://arxiv.org/abs/1710.10196, 2017

CycleGAN
Jun-Yan Zhu Taesung Park Phillip Isola Alexei A. Efros
« Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks »,
https://arxiv.org/abs/1703.10593, 2017





GANs, GANs, more GANs...

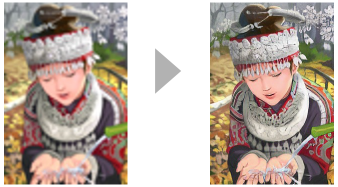
Text-to-Image Synthesis

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas,
« StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks »,
https://arxiv.org/abs/1612.03242, 2016

« This smaller brown bird has white stripes on the coverts, wingbars and secondaries »

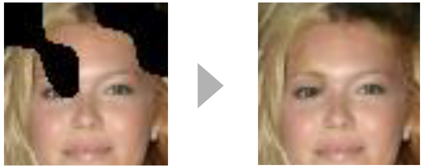
Super-Resolution

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi,
« Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network »,
https://arxiv.org/abs/1609.04802, 2016



Semantic Image Inpainting

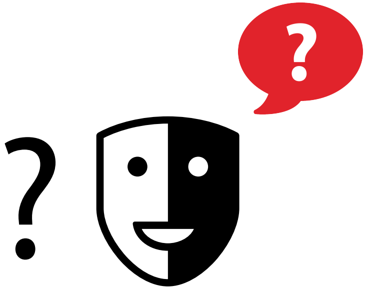
Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Do,
« Semantic Image Inpainting with Deep Generative Models »,
https://arxiv.org/abs/1607.07539, 2016



Face Frontal View Generation

Rui Huang, Shu Zhang, Tianyu Li, Ran He,
« Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis »,
https://arxiv.org/abs/1704.04086, 2017

...and then ?



Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., et al.
« The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation. »
https://doi.org/10.17863/CAM.22520 , 2018

...and then ?

2023

« Réponse Photo », couvertures des n°356 (février 2023) et 357 (mars 2023) - https://www.reponsesphoto.fr/

...and then ?

2023

AE → VAE → GAN → **Diffusion Model**

« Réponse Photo », couvertures des n°356 (février 2023) et 357 (mars 2023) - https://www.reponsesphoto.fr/

FIDLE

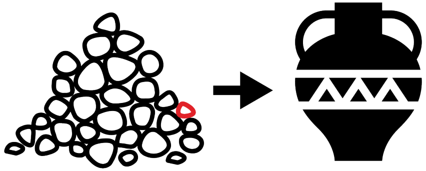
https://youtube.com/@CNRS-FIDLE
https://fidle.cnrs.fr

Jeudi
28
Mars
à 14h00

Next, on Fidle :

L'IA comme un outil

14



Diffusion Model

Text to image

Réalisation, technique et mise en scène :
Agathe, Baptiste et Yanis - UGA/DAPI
Thibaut, Kamel, Léo - IDRIS

Présentions « Flash » et/ou posters ?

jdls@idris.fr