SAISON 22/23

Formation

Introduction au Deep Learning

Generative Adversarial Networks (GANs)

FIDLE

v2.12

Cette session va être enregistrée.
Retrouvez-nous sur notre chaine YouTube :-)

This session will be recorded.
Find us on our YouTube channel :-)

https://fidle.cnrs.fr/youtube

SAISON 22/23

Formation

Introduction au Deep Learning

Generative Adversarial Networks (GANs)

FIDLE

v2.12

Resources

https://fidle.cnrs.fr

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

Course materials (pdf)

Practical work environment*

Corrected notebooks

Videos (YouTube)

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

Resources

You can also subscribe to :

http://fidle.cnrs.fr/listeinfo
Fidle information list

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

Program

FIDLE

20 Séquences
du 17 novembre
au 14 mai 2023

SAISON 22/23

1

History, Fundamental Concepts


Basic Regression
DNN


Basic Classification
DNN

2 3

Hight Dimensionnal Data
CNN

4

Demystify **mathematics** for neural networks.

5

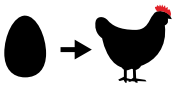
Training strategies
Evaluation


Sparse data (text)
Embedding

6

Sequences data
RNN

7
PyTorch
A small detour with **PyTorch**.

8

«Attention is All You Need»
Transformers

9 New !

Graph Neural Network
GNN

10

Autoencoder networks
AE

11

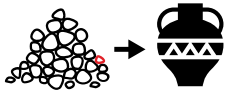
Variational Antoencoder
VAE

12
JDLS 12 mai

Project session
«My project in 180 s»

13

Generative Adversarial Networks
GAN

14 New !

Diffusion Model
Text to image

15

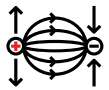
AI, Law, Society and **Ethics**

16 New !

Model and training optimization
Resource efficiency

17

Jean-Zay
GPU acceleration

18 New !

Physics-Informed Neural Networks
PINNS

19

Deep Reinforcement Learning
RL

20

What will be tomorrow's AI
Review & perspectives !

Roadmap

13

Generative Adversarial Networks

GAN

1 About GAN

- Principles
- Training

2 Example 1 : Sheep1

- 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 : Sheep2

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

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

Roadmap

13

Generative Adversarial Networks

GAN

1 **About GAN**
- Principles
- Training

2 **Example 1 : Sheep1**
- Draw me a sheep !

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

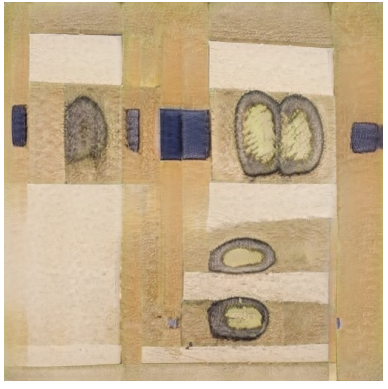
4 **Example 2 : Sheep2**
- Again, but with a WGAN-GP :-)

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

About GAN








https://thisxdoesnotexist.com/

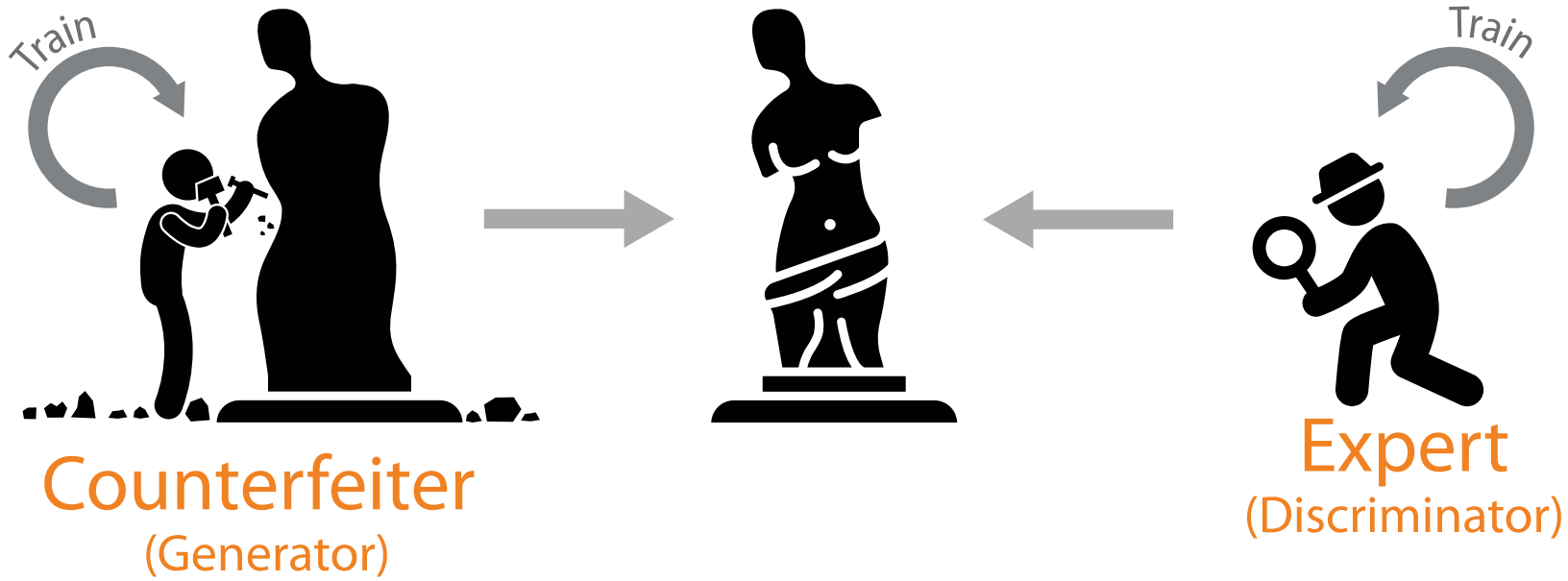
Which Face Is Real ?

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

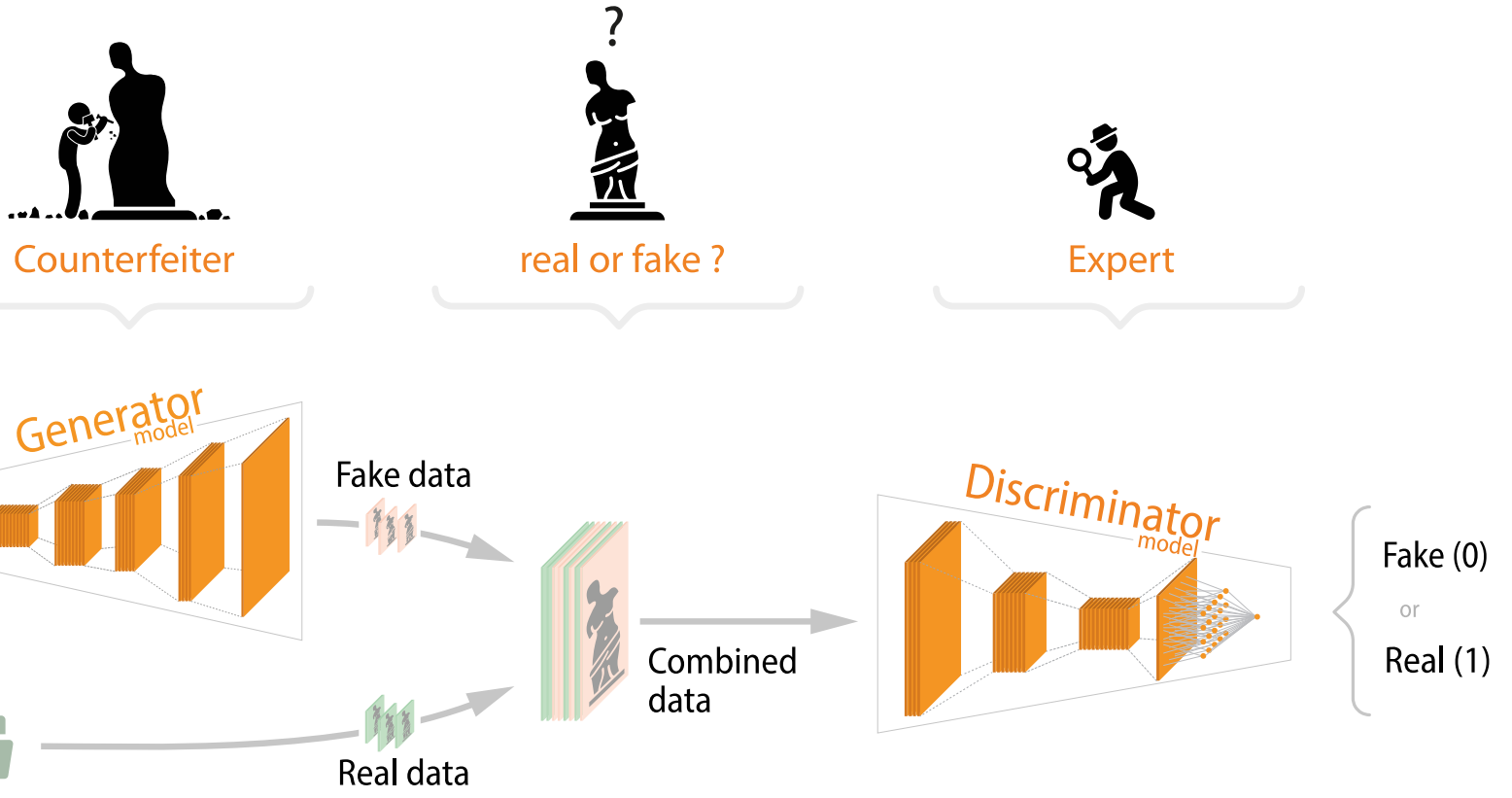
« (…) synthetically generated faces are more trustworthy than real faces. This may be because synthesized faces tend to look more like average faces which themselves are deemed more trustworthy. (…) »

About GAN

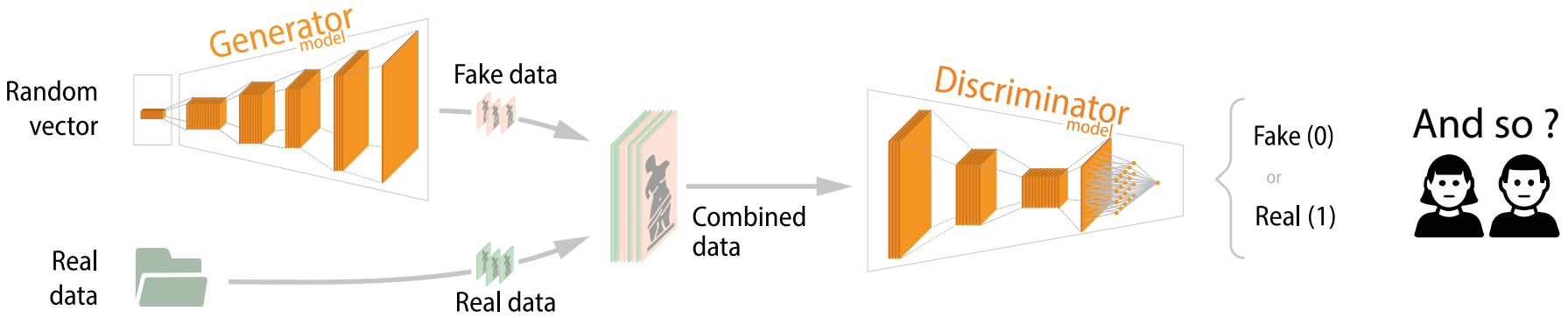


Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville et Yoshua Bengio, « Generative Adversarial Networks », in Advances in Neural Information Processing Systems 27, 2014
https://arxiv.org/abs/1406.2661

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:

n epochs

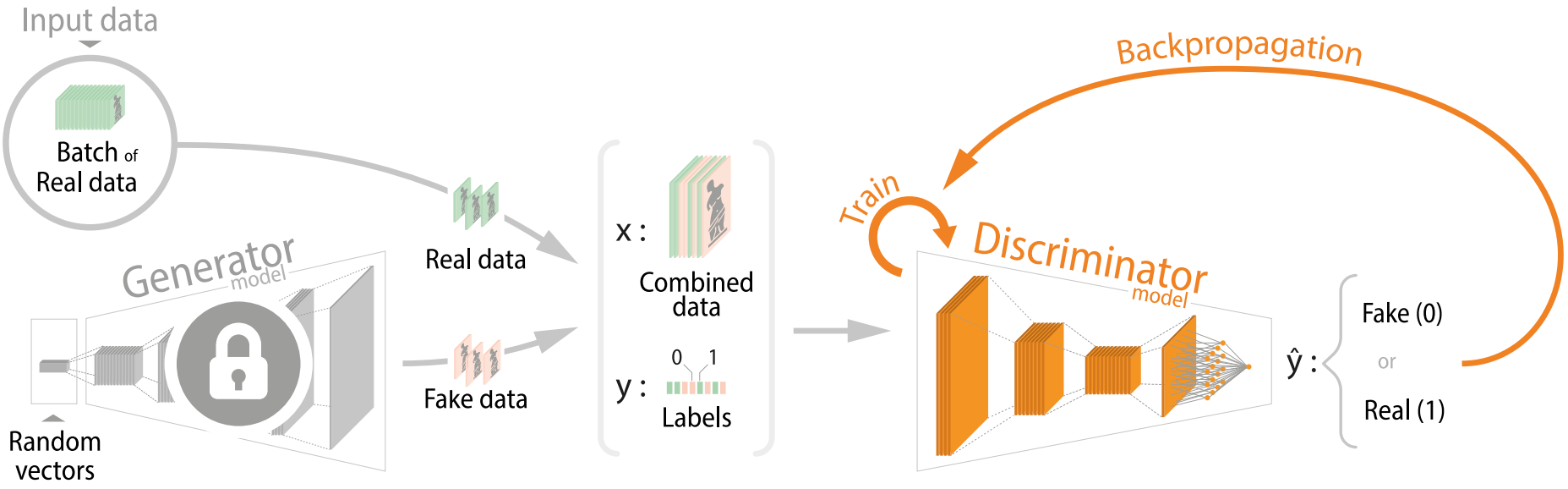
Step 1 - Update Discriminator

Step 2 - Update Generator

How to train a GAN ?

Step1 - Update Discriminator

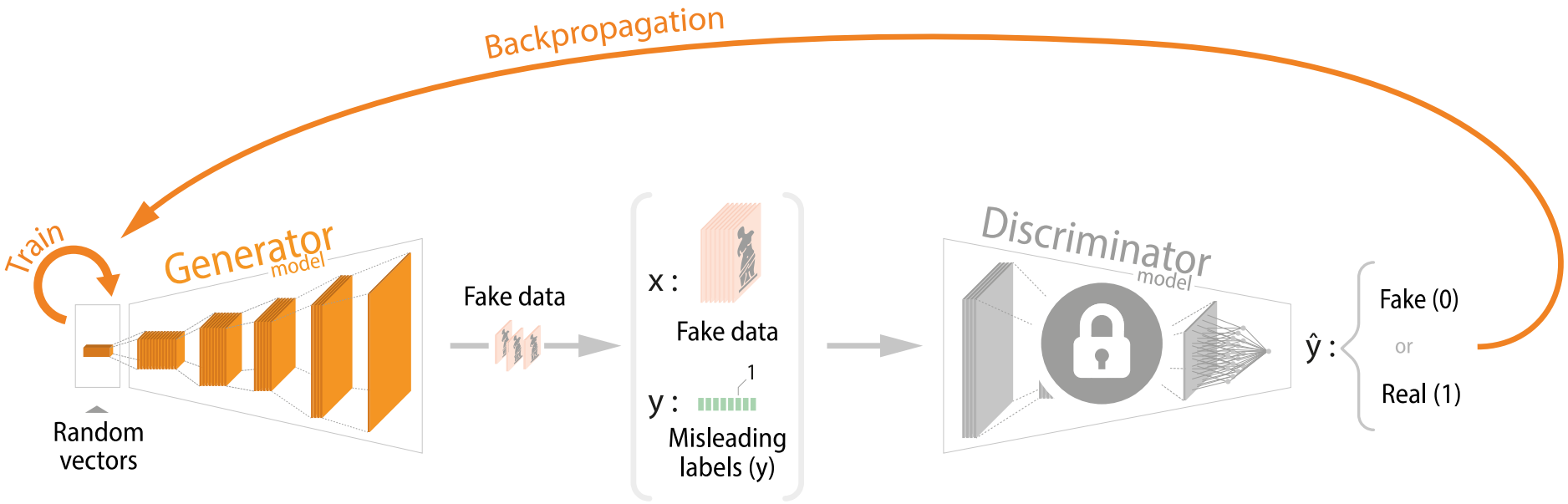
Generator is locked



How to train a GAN ?

Step2 - Update Generator

Discriminator is locked



Roadmap

13

Generative Adversarial Networks

GAN

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

2. **Example 1 : Sheep1**
   - Draw me a sheep !

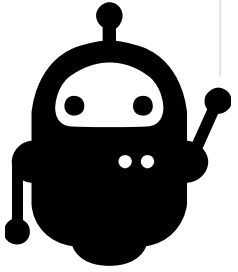
3. **From GANs to *GANs**
   - Wasserstein and Earth Moving Distance
   - From GAN to WGAN
   - From WGAN to WGAN-GP

4. **Example 2 : Sheep2**
   - Again, but with a WGAN-GP :-)

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

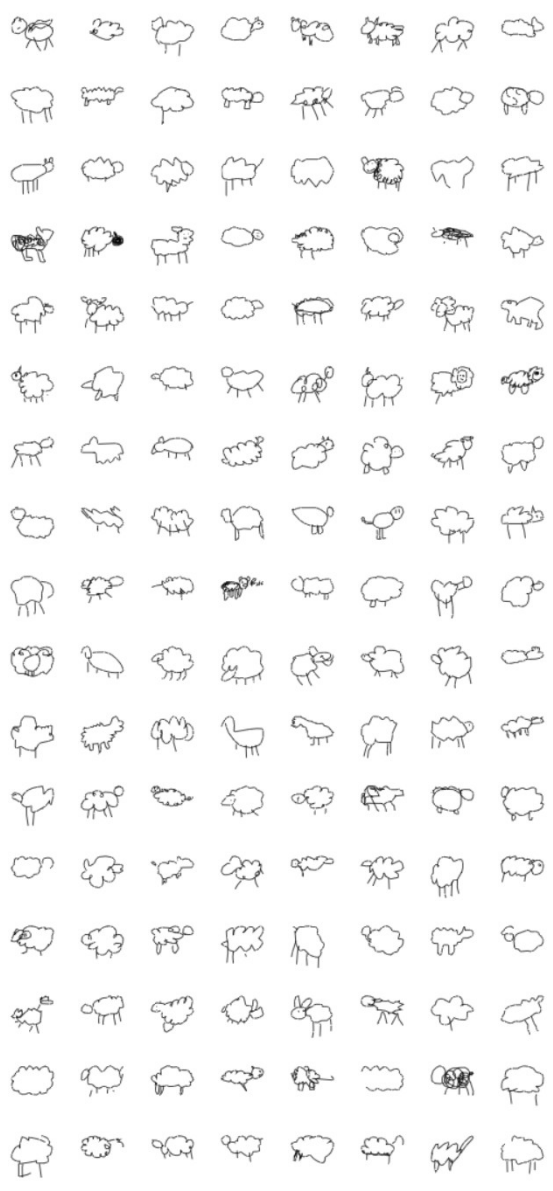
Draw me a sheep !

Notebook : [SHEEP1]



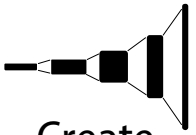
Objective :
Learn to draw or generate sheep.

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



SHEEP1

6m on a V100 !
(10 epochs)



Import and init

Load dataset & have a look !

Create Generator

History

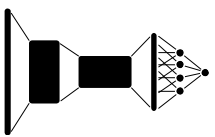
START

Step 1 → Step 2 → Step 3 → Step 4 → Step 5 → Step 6 → Step 7 → Step 8 → END

Parameters

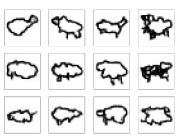
```
latent_dim = 128
Scale      = 0.01
```

Create Discriminator



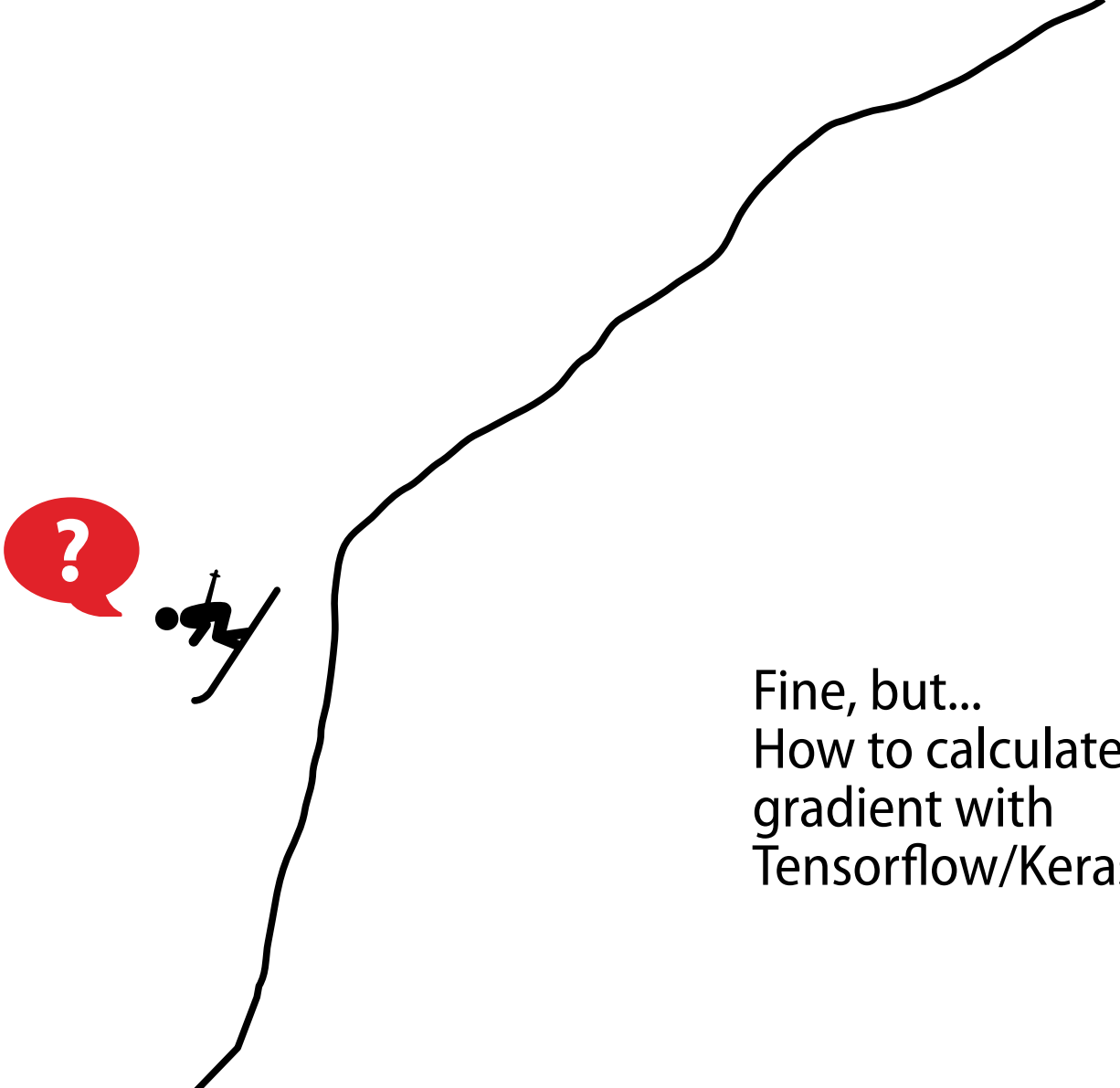
Build, compile & train

Drawing new sheep !



Objectives :

Learn to draw sheep !



Fine, but...
How to calculate a gradient with Tensorflow/Keras ?

How to calculate its preferred gradient?

```
import numpy as np
import tensorflow as tf

# ---- 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))

x = tf.Variable(2.0)
print('f(x) is : ', f(x))
```

```
f(1) is :  0
f(2) is :  7
f(x) is :  tf.Tensor(7.0, shape=(), dtype=float32)
```

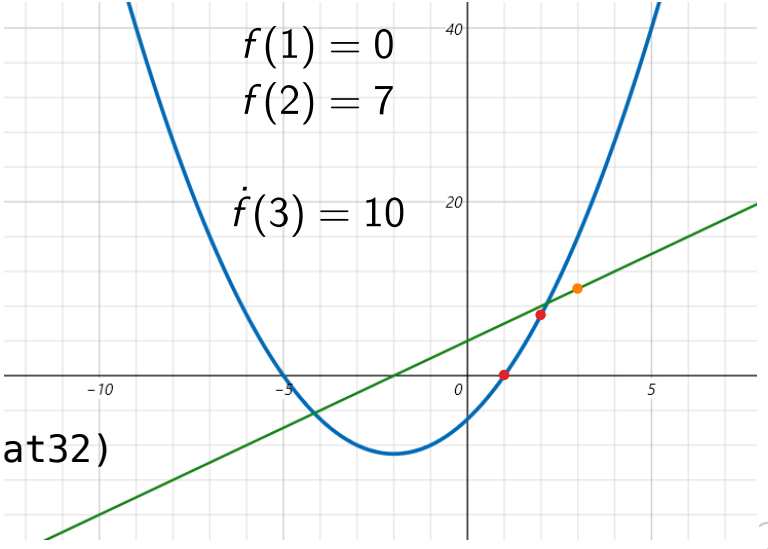
$$\begin{cases} f(x) = x^2 + 4x - 5 \\ \dot{f}(x) = \dfrac{\delta f}{\delta x} = 2x + 4 \end{cases}$$

We have :

$f(1) = 0$

$f(2) = 7$

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



How to calculate its preferred gradient?

```
# ---- Derivative using tf
#

with tf.GradientTape() as tape:
    x = tf.Variable(3.0)
    y = f(x)

dy = tape.gradient(y, x)  # dy/dx

print(dy)
```

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

We have :

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

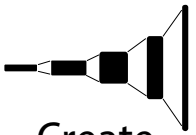
$$\dot{f}(3) = 10$$

```
tf.Tensor(10.0, shape=(), dtype=float32)
```

More about Automatic Differentiation and Gradients :
https://www.tensorflow.org/guide/autodiff
https://www.tensorflow.org/api_docs/python/tf/GradientTape

SHEEP1

6m on a V100 !
(10 epochs)



Import and init

Load dataset & have a look !

Create Generator

History

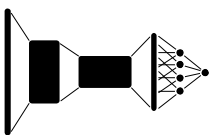
START

Step 1 → Step 2 → Step 3 → Step 4 → Step 5 → Step 6 → Step 7 → Step 8 → END

Parameters

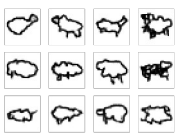
```
latent_dim = 128
Scale      = 0.01
```

Create Discriminator



Build, compile & train

Drawing new sheep !



Objectives :

Learn to draw sheep !

Roadmap

13

Generative Adversarial Networks

GAN

1 **About GAN**

- Principles
- Training

2 **Example 1 : Sheep1**

- 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 : Sheep2**

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

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

Original GAN

2014

→ Original paper [1] :

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

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

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.

Categorical crossentropy :

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

Discrimator 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 : fake : 0, real : 1

[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

Original GAN

2014

→ Original paper [1] :

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

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

GAN loss functions

Limitations :

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

Categorical crossentropy :

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

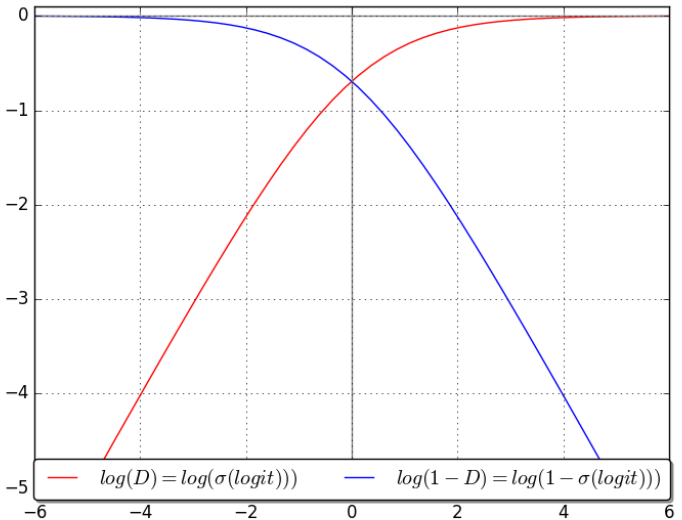


Figure 6: Illustration of vanishing gradient in the negative quadrant when using the original loss formulation log(1-D) (blue curve). The x-axis is D's logit (output of last layer before sigmoid activation). Minimising log(1-D) yields the same solution as maximising log(D) (red curve) but the red curve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

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

[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 [2]:

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

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

Optimization problem [3]

(*) Distance du terrassier

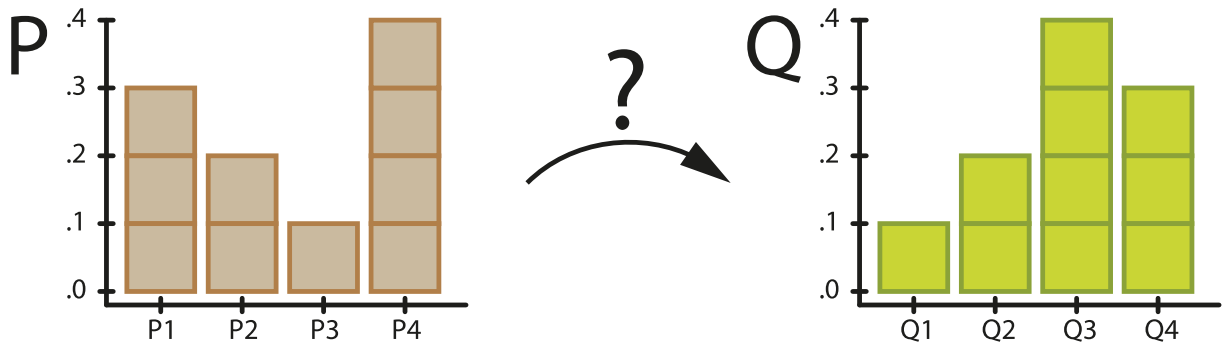
[2] Martin Arjovsky, Soumith Chintala, Léon Bottou, « Wasserstein GAN », https://arxiv.org/abs/1701.07875, 2017

[3] 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:

$$W = \text{Number of boxes moved} \times \text{Moving distance of the boxes}$$

Wasserstein-1 Earth-Mover Distance

We can write :

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

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

$\delta_0 = 0$

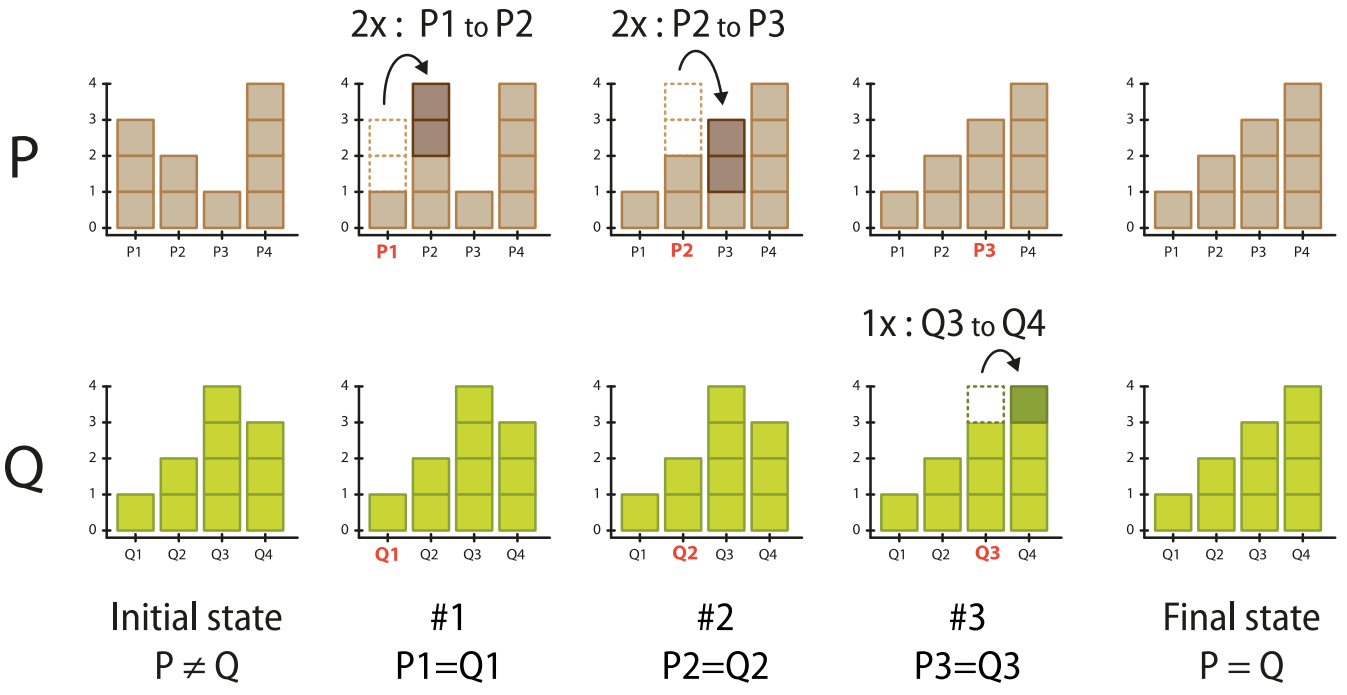
$\delta_1 = 0 + 3\text{-}1 = 2$

$\delta_2 = 2 + 2\text{-}2 = 2$

$\delta_3 = 2 + 1\text{-}4 = -1$

$\delta_4 = -1 + 4\text{-}3 = 0$

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



Wasserstein-1 Earth-Mover Distance

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

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

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

A little more on the subject:

[4] Lilian Weng, « From GAN to WGAN », https://arxiv.org/abs/1904.08994, 2019

[5] Ludovic Platon , « GAN : Vers une meilleure estimation des distributions » (fr), https://www.aquiladata.fr/insights/gan-vers-une-meilleure-estimation-des-distributions/

Wasserstein GAN

2017, dec .6

→ Wasserstein GAN [2] :

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

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

WGAN loss functions

$x^{(i)}$	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.

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 $L_{\text{generator}}$

Note : { fake : low score ; real : hight score }

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

Wasserstein GAN

2017, dec .6

→ Wasserstein GAN [2] :

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

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

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 $L_{\text{generator}}$

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

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 : { fake : low score ; real : hight score }

[2] 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

2017, dec .6

→ Wasserstein GAN [2] :

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

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

WGAN loss functions

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

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]

Note : { fake : low score ; real : hight score

[2] 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 - GP

2017, dec 25

→ Wasserstein GAN with Gradient Penalty [6] :

Here is the gradient penalty

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

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

WGAN-GP loss functions

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 $L_{\text{generator}}$

$$\hat{x} = \varepsilon x + (1-\varepsilon)\tilde{x}$$
with $t \sim U[0,1]$

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.

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

[6] 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

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

Roadmap

13

Generative Adversarial Networks

GAN

1 **About GAN**

- Principles
- Training

2 **Example 1 : Sheep1**

- Draw me a sheep !

3 **From GANs to *GANs**

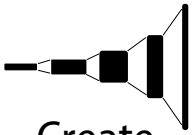
- Wasserstein and Earth Moving Distance
- From GAN to WGAN
- From WGAN to WGAN-GP

4 **Example 2 : Sheep2**

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

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

6m on a V100 !
(10 epochs)



START

Import and init

Load dataset & have a look !

Create Generator

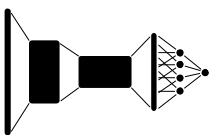
History

Step 1 → Step 2 → Step 3 → Step 4 → Step 5 → Step 6 → Step 7 → Step 8 → END

Parameters

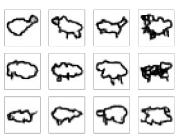
```
latent_dim = 128
Scale      = 0.01
…
n_critic = 2
```

Create Discriminator



Build, compile & train

Drawing new sheep !



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

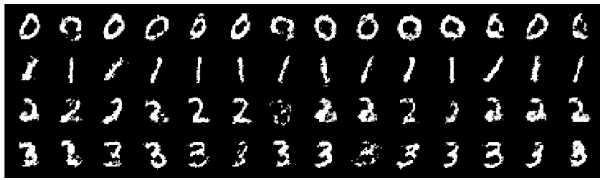
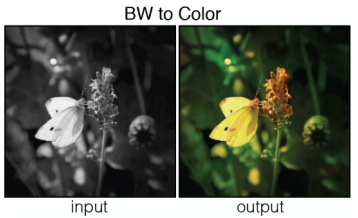


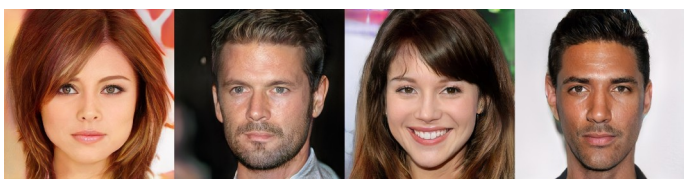
Image-to-Image Translation

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...

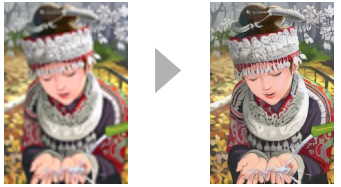
Text-to-Image Synthesis

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 »

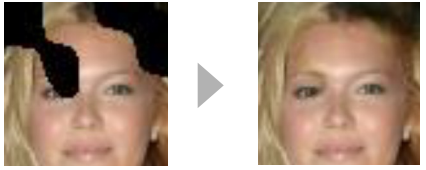
Super-Resolution

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



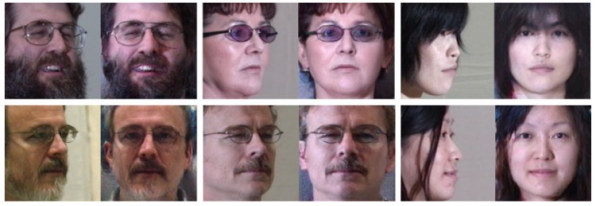
Semantic Image Inpainting

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 ?

Séquence 14 - **AI, droit, société et éthique**
Jeudi 23 mars, 14h

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 ?

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

...and then ?

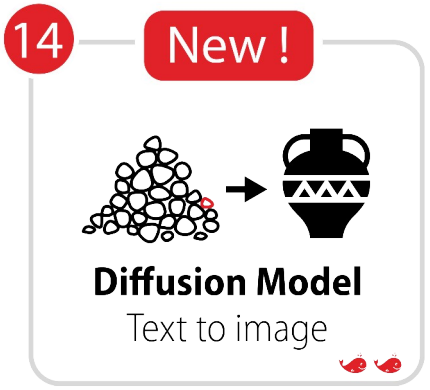
AE → VAE → GAN → Diffusion model

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

New !

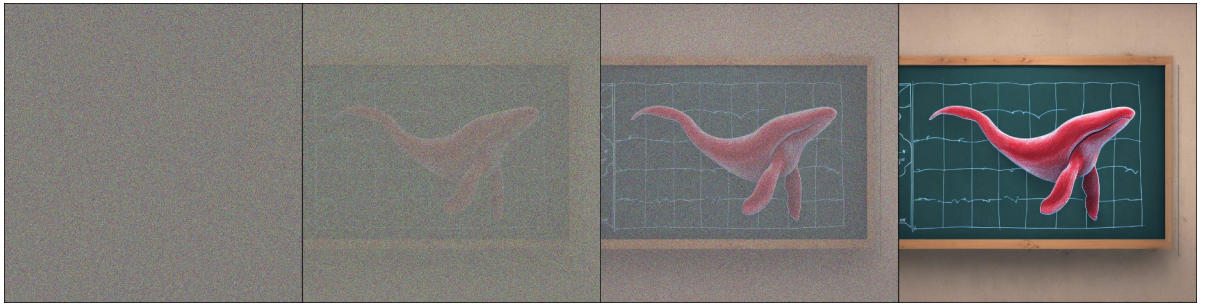
Next, on Fidle :

Jeudi 16 mars,



Épisode 14 :

Diffusion model, text to image



- Principe des Diffusion Model
- Le processus et l'architecture du DDPM
- Améliorations et optimisations des DDPM
- Utilisation des Diffusion Model

Exemple proposé : Génération d'une garde robe :-)

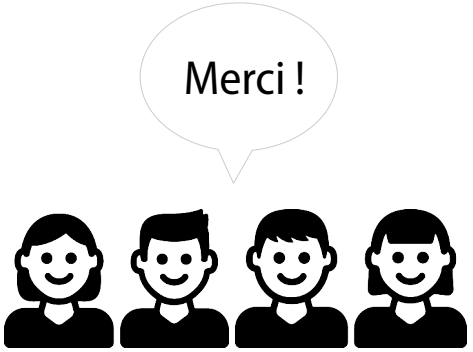
Durée : 2h

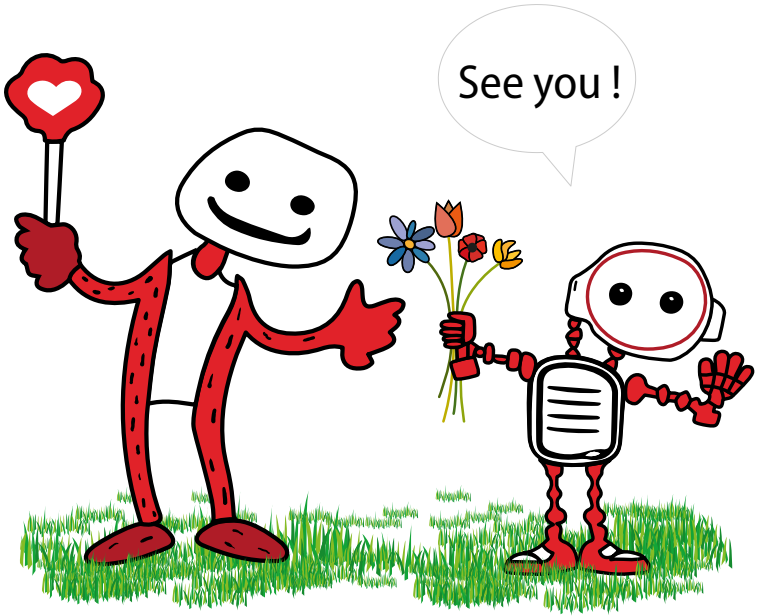
Next on Fidle :

Jeudi 16 mars,

Séquence 9 :

Diffusion model, text to image





« A funny person who says hello with his harm »
« A funny robot with a big brain who says hello with his harm. »
by Stable Diffusion 2.1-768

To be continued...

Contact@fidle.cnrs.fr
FIDLE https://fidle.cnrs.fr
YouTube https://fidle.cnrs.fr/youtube

FIDLE