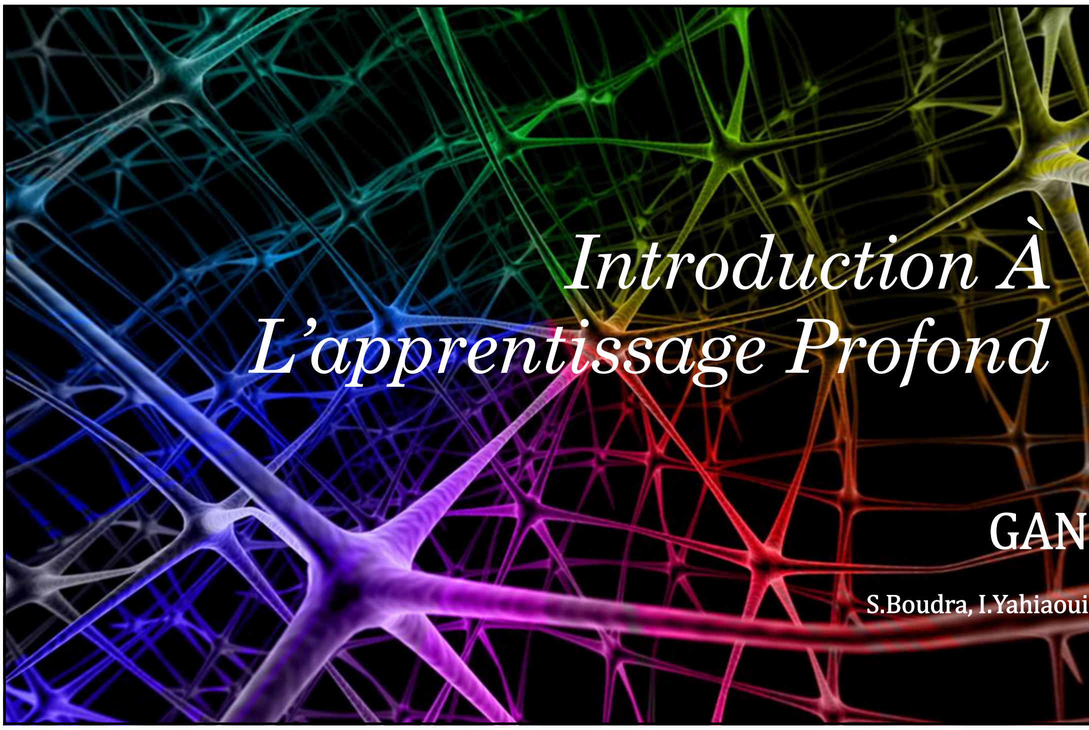


Introduction À L'apprentissage Profond

GAN

S.Boudra, I.Yahiaoui

1

1. CNN
2. Quelques Architectures CNN
3. Data Augmentation
4. Transfer Learning

Rappel

2

Generative Adversial Networks: (GAN)

3

GAN:quoi?

- Des modèles génératifs qui utilisent des techniques d'apprentissage profond
- Une méthode non-supervisée générative:
  - Les données ne sont pas labelisées ni catégorisées.
  - Pas de généralisation pour la prédiction des classes de nouvelles données.
- Un GAN découvre automatiquement des motifs (pattern) à partir d'un ensemble de données, et est capable de générer des données vraisemblables.

4

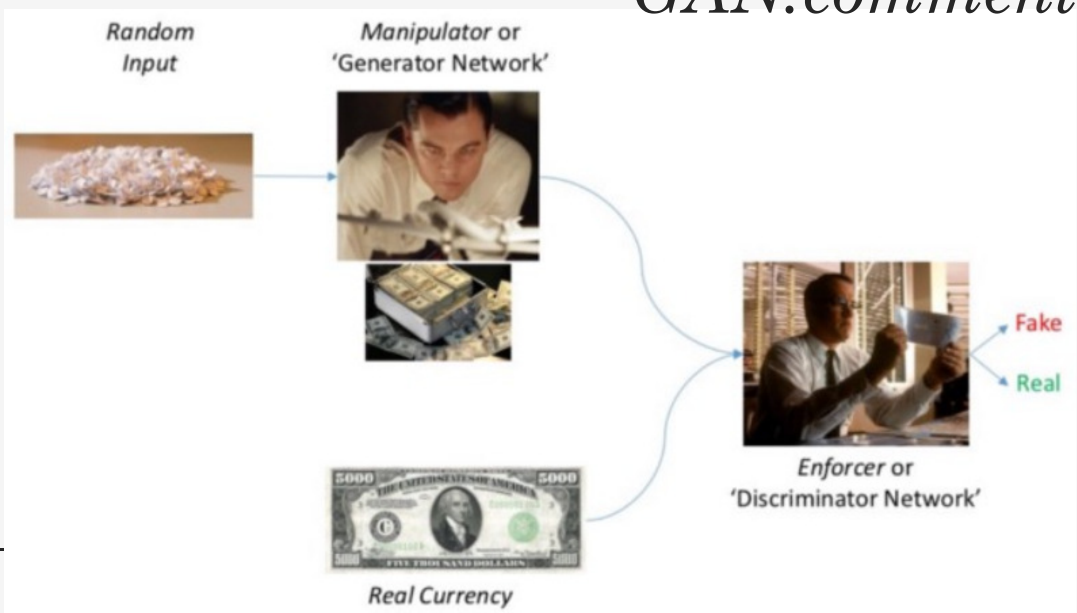
GAN:comment?

- GAN est formalisé comme un problème supervisé pour réaliser une tache non supervisée:
- Composé d'une paire de réseaux de neurones: generator *G*/ discriminator *D* :
  - *G* ***produit*** de nouvelles images et
  - *D* essaye de ***distinguer*** les images générées des réelles. (partie supervisée)

- Les deux modèles sont entrainés ensemble de façon antagoniste à somme nulle.
- *G* essaye de tromper *D*, et *D* essaye de rester discriminant.

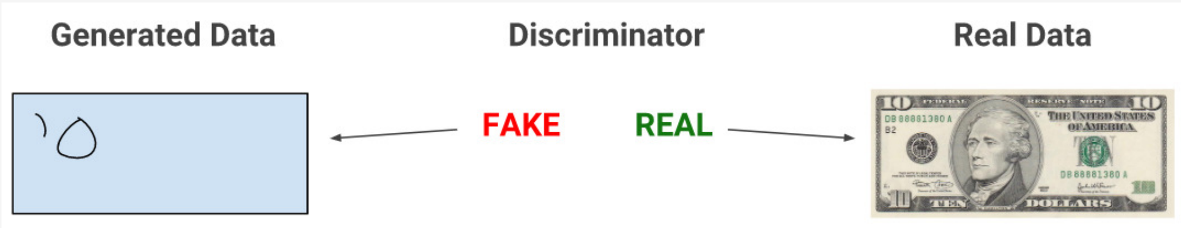
5

GAN:comment?



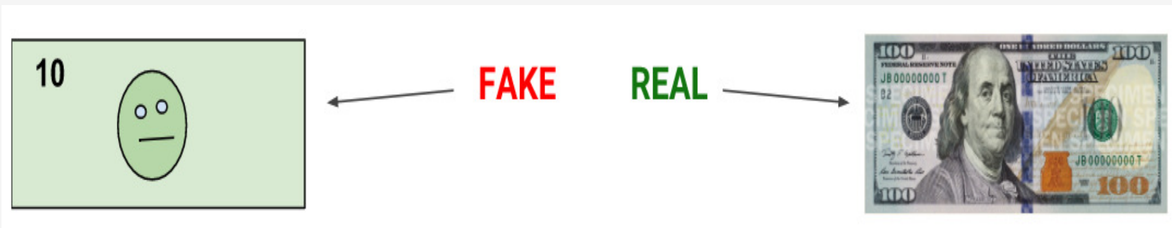
6

GAN:comment?



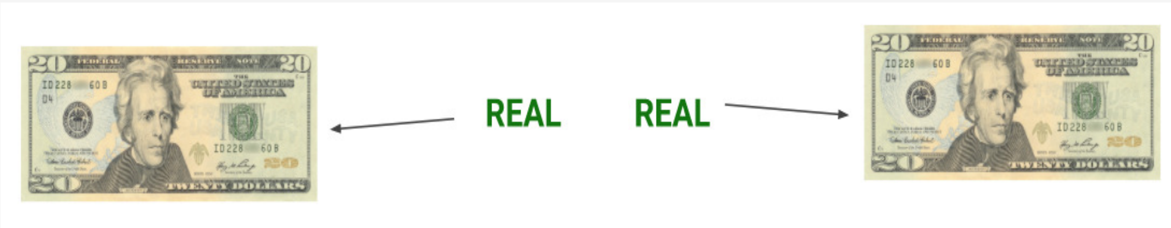
7

GAN:comment?



8

GAN:comment?



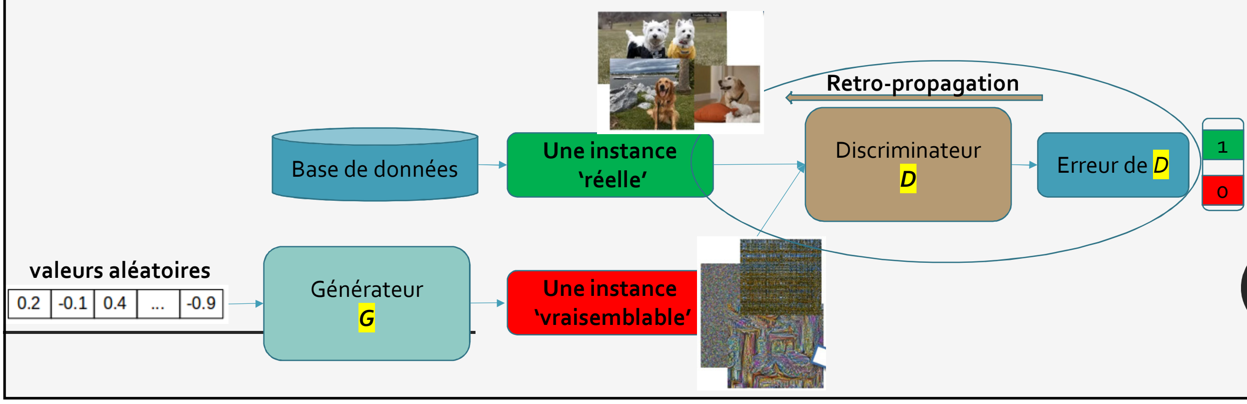
9

GAN:détails

10

GAN: descriminator (D)

- <u>Le discriminateur *D*</u> (classifieur binaire)
- Les données pour <u>entrainer</u> le discriminateur proviennent de:
  - ➢ **Données réelles:** de la base de données, annotées (1)
  - ➢ **Données vraisemblables:** crées par le générateur, annotées (0)
- Calcul de l'erreur de *D* et mise à jour de ses poids par retro-propagation.

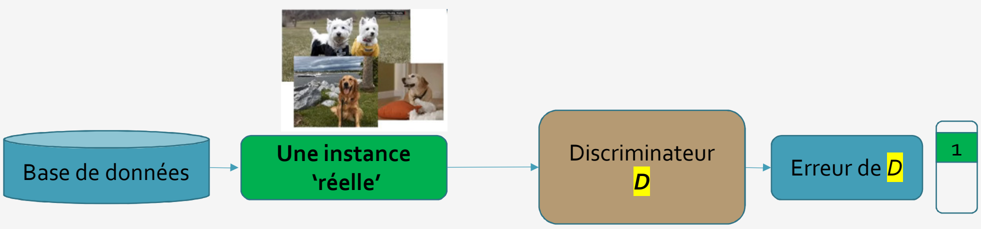


11

GAN: descriminator loss

- Le Discriminateur D veut:
  - ➢ Reconnaitre une vraie image x comme 'vraie' => valeur prédite proche de 1

Label =1 valeur erreur grande
Pred = 0.1



12

GAN: descriminator loss

- Le Discriminateur D veut:
  - ➢ Reconnaitre une vraie image x comme 'vraie' => valeur prédite proche de 1

Label =1 valeur erreur faible
Pred = 0.9

Base de données → Une instance 'réelle' → Discriminateur *D* → Erreur de *D* → 1

13

GAN: descriminator loss

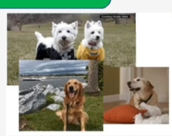
- Le Discriminateur D veut:
  - ➢ Reconnaitre une vraie image x comme 'vraie' => valeur prédite proche de 1

Label =1 valeur erreur grande -ln(0.1) = 2.3
Pred = 0.1

L= -ln(Pred)

Label =1 valeur erreur faible -ln(0.9) = 0.1
Pred = 0.9

Base de données → Une instance 'réelle' → Discriminateur *D* → Erreur de *D* → 1

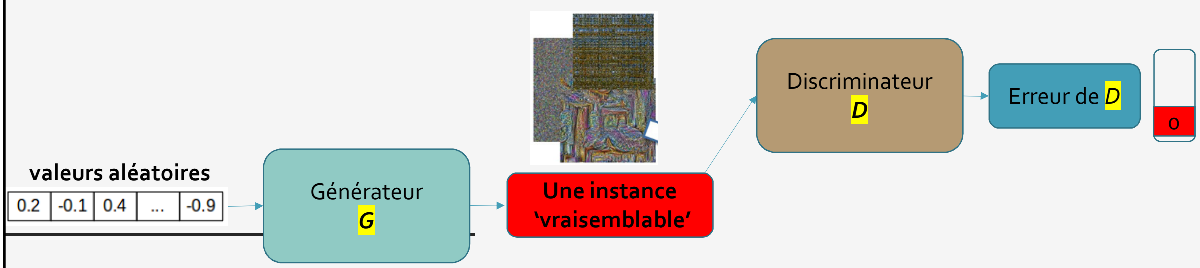


14

GAN: descriminator loss

- Le Discriminateur D veut:
  - ➢ Reconnaitre une fausse image x comme 'fausse' => valeur prédite proche de 0

Label =0 valeur erreur faible
Pred = 0.1

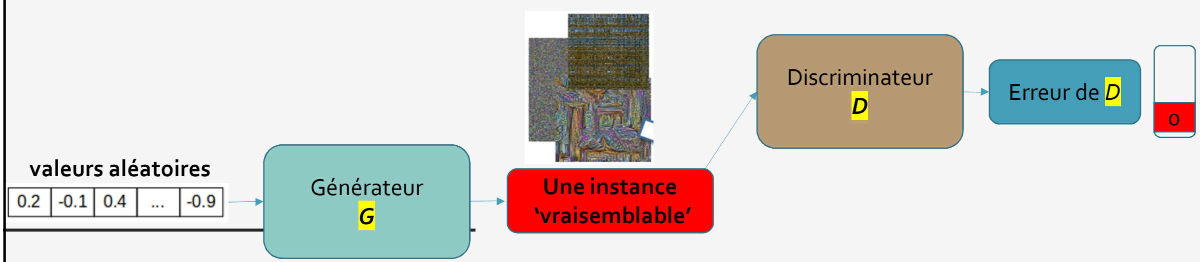


15

GAN: descriminator loss

- Le Discriminateur D veut:
  - ➢ Reconnaitre une fausse image x comme 'fausse' => valeur prédite proche de 0

Label =0 valeur erreur grande
Pred = 0.9



16

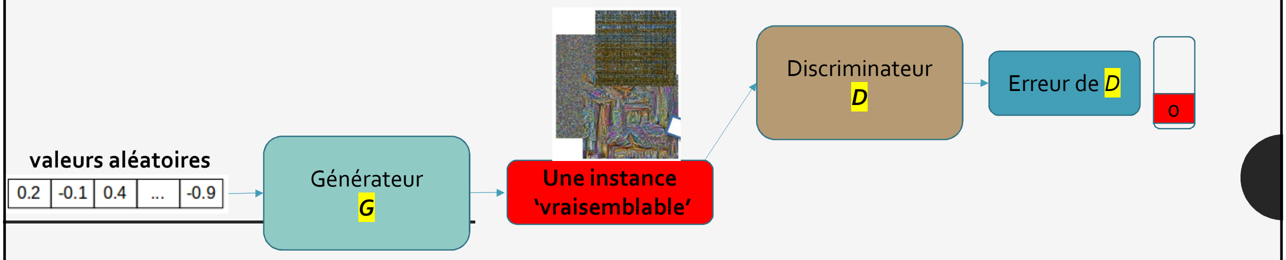
GAN: descriminator loss

- Le Discriminateur D veut:
  - ➢ Reconnaitre une fausse image x comme 'fausse' => valeur prédite proche de 0

Label =0 valeur erreur faible -ln(1-0.1)= -ln(0.9)= 0.1
Pred = 0.1

L= -ln(1-Pred)

Label =0 valeur erreur grande -ln(1-0.9) = -ln(0.1) = 2.3
Pred = 0.9

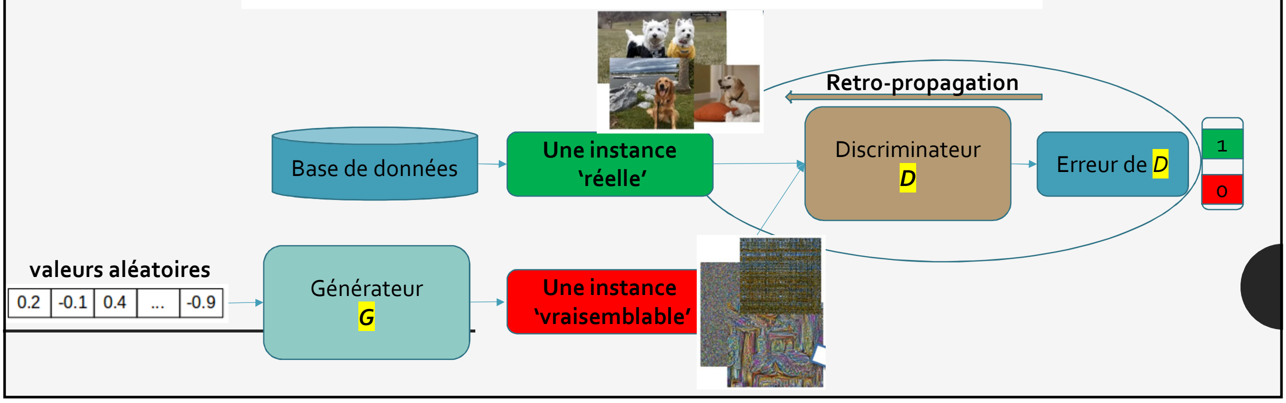


17

GAN: descriminator (D)

- Descriminator-loss

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



18

GAN: generator (G)

- **Un générateur *G*:** apprend à générer des données vraisemblables à partir d'un vecteurs de valeurs aléatoires.
- Calculer la fonction d'erreur du discriminateur.
- Retro-propager à travers le discriminateur et le générateur.
- Mettre à jours les poids du générateur G **uniquement.** Les poids de D restent inchangés.

19

GAN: generator (G)

- Le Générateur G veut:
  - ➢ Générer de fausses images G(z) que le discriminateur prend pour réelles.

  Label =1 valeur erreur grande

  Pred = 0.1

20

GAN: generator (G)

- Le Générateur G veut:
  - ➢ Générer de fausses images G(z) que le discriminateur prend pour réelles.

Label =1 valeur erreur faible
Pred = 0.9

21

GAN: generator (G)

- Le Générateur G veut:
  - ➢ Générer de fausses images G(z) que le discriminateur prend pour réelles.

Label =1 valeur erreur grande -ln(0.1) = 2.3
Pred = 0.1

L= -ln(Pred)

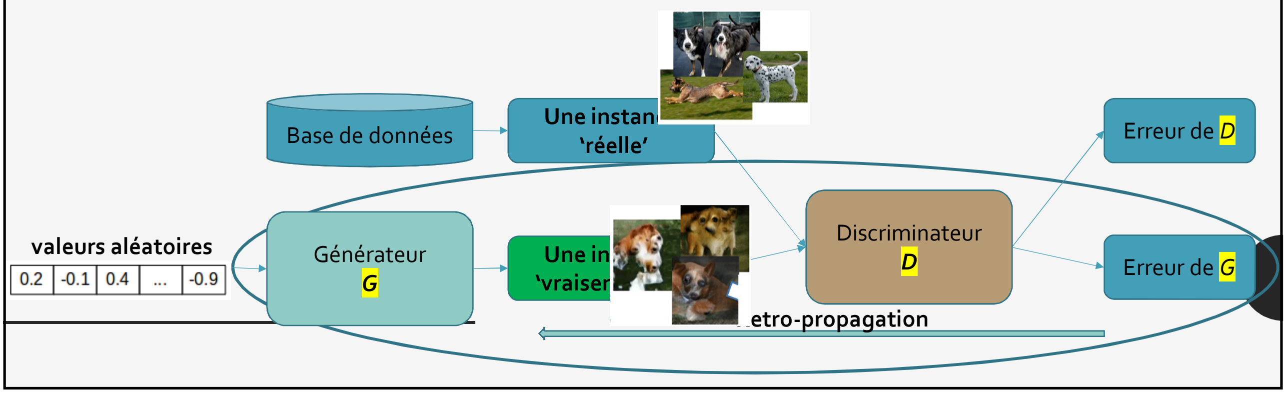
Label =1 valeur erreur faible -ln(0.9) = 0.1
Pred = 0.9

22

GAN: generator (G)

- Generator-loss : $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)$

- Au lieu d'entrainer G à minimiser **log(1-D(G(z)))**, on l'entraine à maximiser **log(D(G(z))**.



23

GAN: zero-sum game

Score de sortie pour une vraie image

Score de sortie pour une image produite par G

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- Le Discriminateur D veut:
  - ➢ Reconnaitre une vraie image x comme 'vraie' => un score proche de 1
  - ➢ Reconnaitre une fausse image x comme 'fausse' => un score proche de 0
- Le Générateur G veut:
  - ➢ Générer de fausses images G(z) que le discriminateur prend pour réelles.

24

GAN: Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

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

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

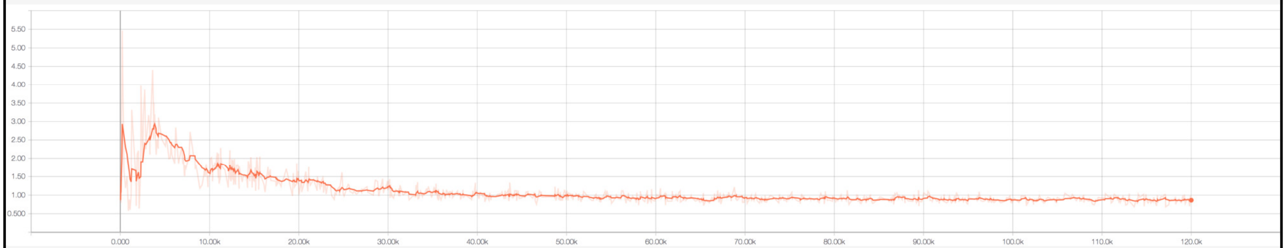
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

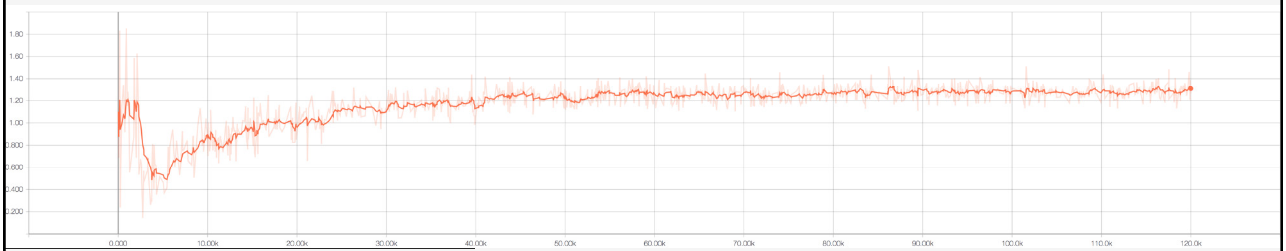
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

25

GAN: Algorithm



generator error



descriminator error

26

GAN: applications

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015).

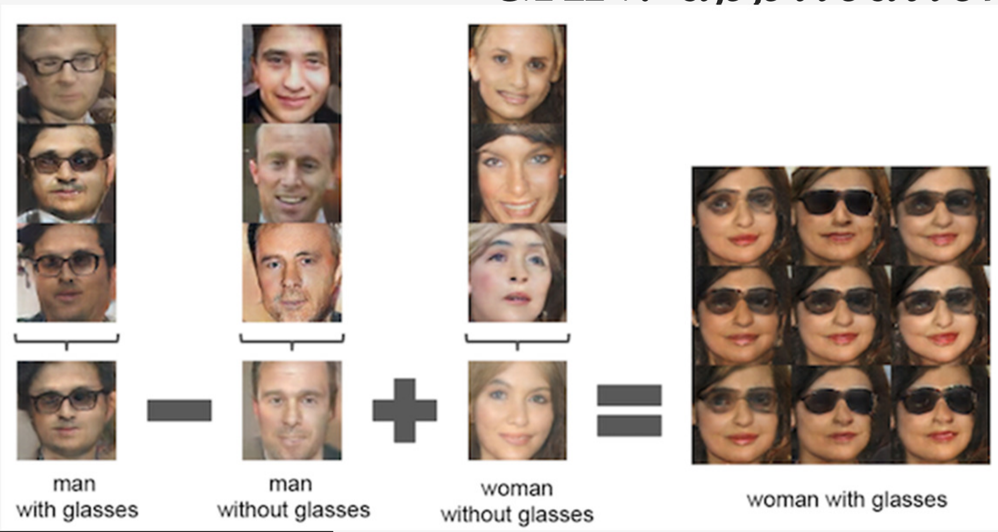
27

GAN: applications: synthetic faces

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." *arXiv preprint arXiv:1710.10196* (2017).

28

GAN: applications



Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015).

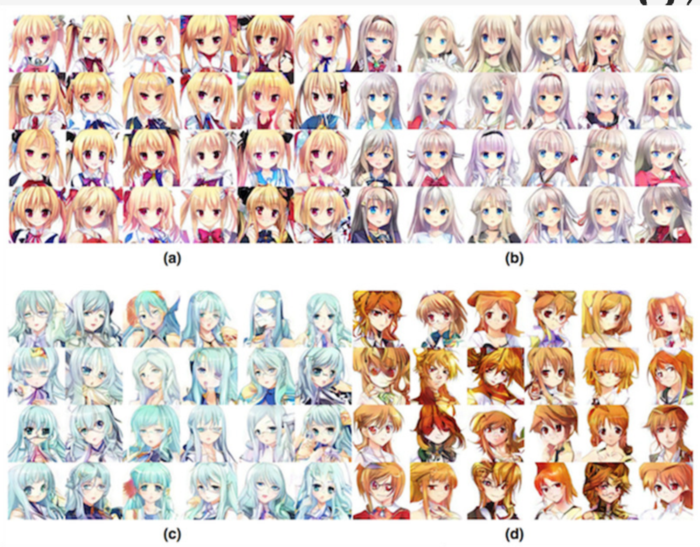
29

GAN: applications: realistic images

Brock, Andrew, Jeff Donahue, and Karen Simonyan. "Large scale gan training for high fidelity natural image synthesis." *arXiv preprint arXiv:1809.11096* (2018).

30

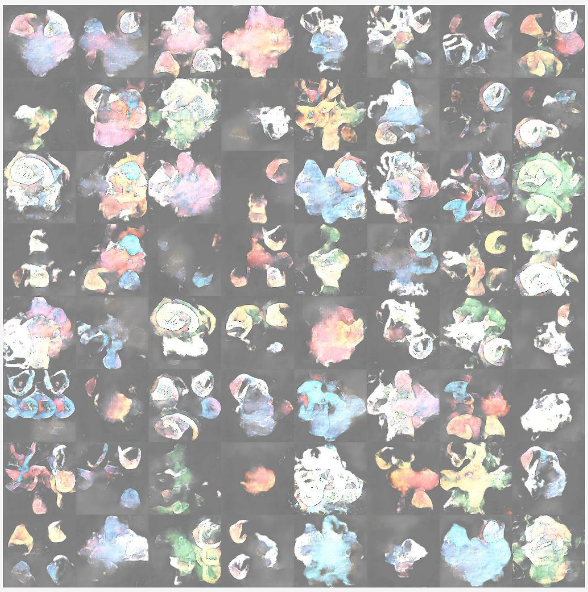
GAN: applications: anime caracters



Jin, Yanghua, et al. "Towards the automatic anime characters creation with generative adversarial networks." *arXiv preprint arXiv:1708.05509* (2017)

31

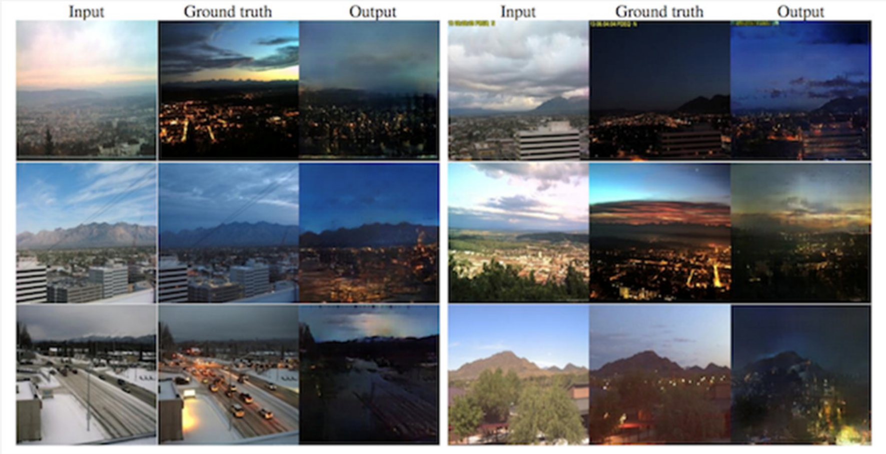
GAN: applications: Project PokeGAN



ttps://awesomeopensource.com/project/moxiegushi/pokeGAN

32

GAN: applications: image-to-image translation



Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017.

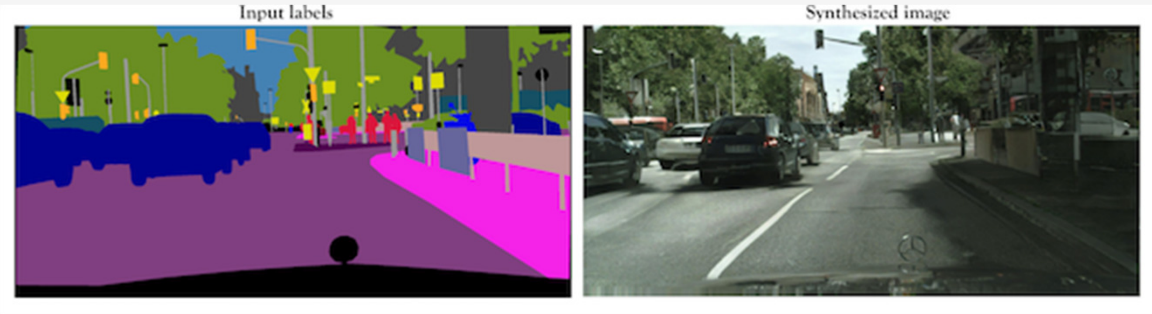
33

GAN: applications: image-to-image translation

Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017.

34

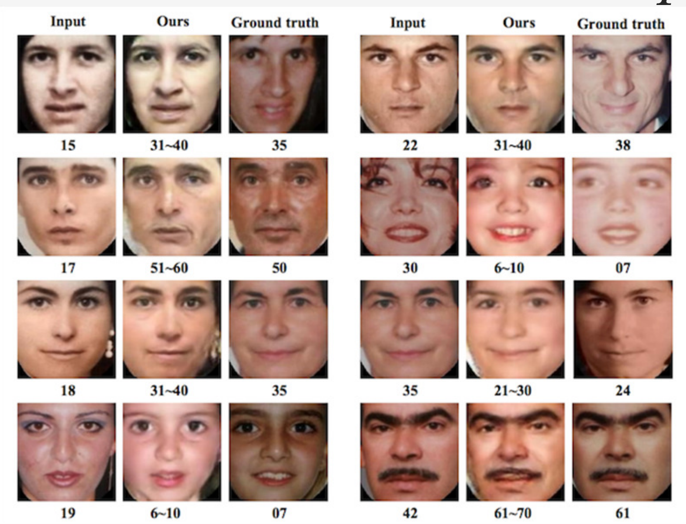
GAN: applications: semantic-image-to-photo translation



Wang, Ting-Chun, et al. "High-resolution image synthesis and semantic manipulation with conditional gans." *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2018.

35

GAN: applications: face aging



Zhang, Zhifei, Yang Song, and Hairong Qi. "Age progression/regression by conditional adversarial autoencoder." *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017.

36

Références

- https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d
- http://scs.ryerson.ca/~aharley/vis/conv/flat.html
- https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721
- http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf
- http://cs231n.github.io/assets/conv-demo/index.html
- http://cs231n.github.io/convolutional-networks/
- https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
- http://neuralnetworksanddeeplearning.com/chap1.html
- http://cedric.cnam.fr/vertigo/Cours/ml2/preambule.html
- http://cs231n.github.io/convolutional-networks/#case
- https://www.learnopencv.com/number-of-parameters-and-tensor-sizes-in-convolutional-neural-network/
- https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848
- https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#a253
- https://www.youtube.com/watch?v=c1RBQzKsDCk&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF&index=17&t=0s&pbjreload=10

37

Références

- http://cs231n.github.io/convolutional-networks/#case
- https://www.learnopencv.com/number-of-parameters-and-tensor-sizes-in-convolutional-neural-network/
- https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848
- https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#a253
- https://www.youtube.com/watch?v=c1RBQzKsDCk&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF&index=17&t=0s&pbjreload=10
- https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/
- https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/
- https://keras.io/preprocessing/image/
- https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a
- http://cs231n.github.io/transfer-learning/
- https://towardsdatascience.com/what-is-deep-transfer-learning-and-why-is-it-becoming-so-popular-91acdcc2717a
- https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a
- http://cedric.cnam.fr/vertigo/Cours/ml2/docs/coursDeep5.pdf

38