Invatare automata in arta vizuala

Invatare nesupervizata
Retele generative

Invatare nesupervizata [1]

- Dataset
  - Doar date
  - Fare anotari
- Scop
  - Descoperea de patternuri / structura in date.
- Exemple
  - Clusterizare
  - Reducere dimensiunii datelor
  - Invatarea de feature-uri
  - Estimarea densitatii
  - Generare

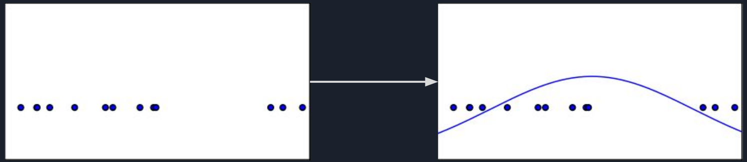
“What I cannot create, I do not understand.”

—Richard Feynman

Invatare nesupervizata [2] - Generare de imagini

- Input
  - Example din lumea reala - esantioane din distributia reala a datelor

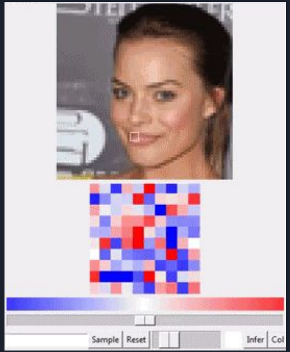
- Modelul generativ
  - Genereaza exemple noi de imagini asemanatoare cu distributia pe care a fost antrenat
  - O retea care genereaza la output imagini - esantioane din distributia modelului
  - Estimarea densitatii



  - Generare de sample-uri

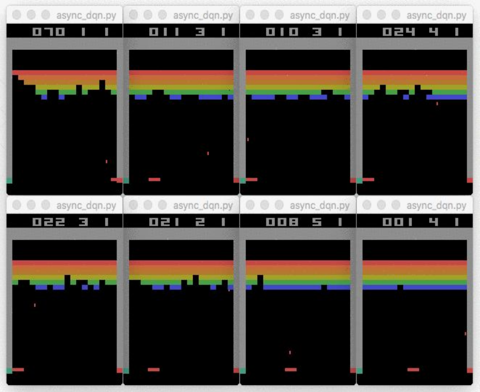
Modele generative

- Sample-uri generate pentru arta
- Super-resolutie
- Colorizare
- Automatizarea generarii de date



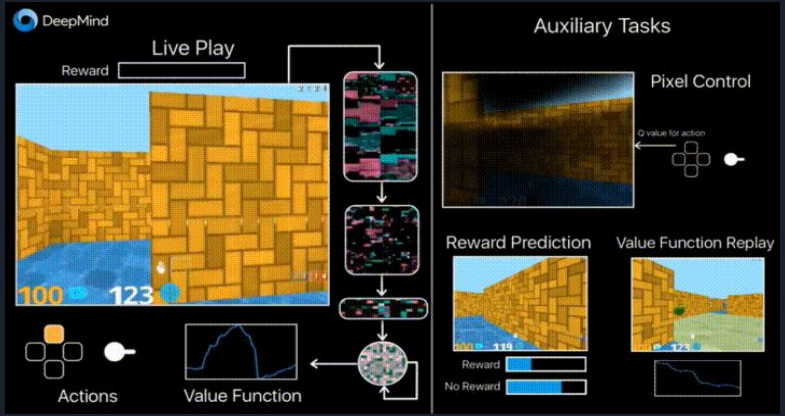
Modele generative

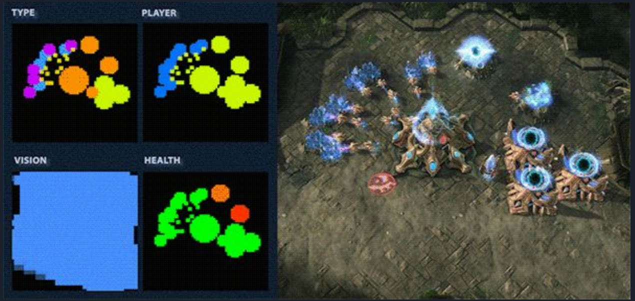
- Modele generative aplicate pe date time-series pot fi folosite pentru simulare si planificare (aplicatii in reinforcement learning)



Modele generative

- Game playing



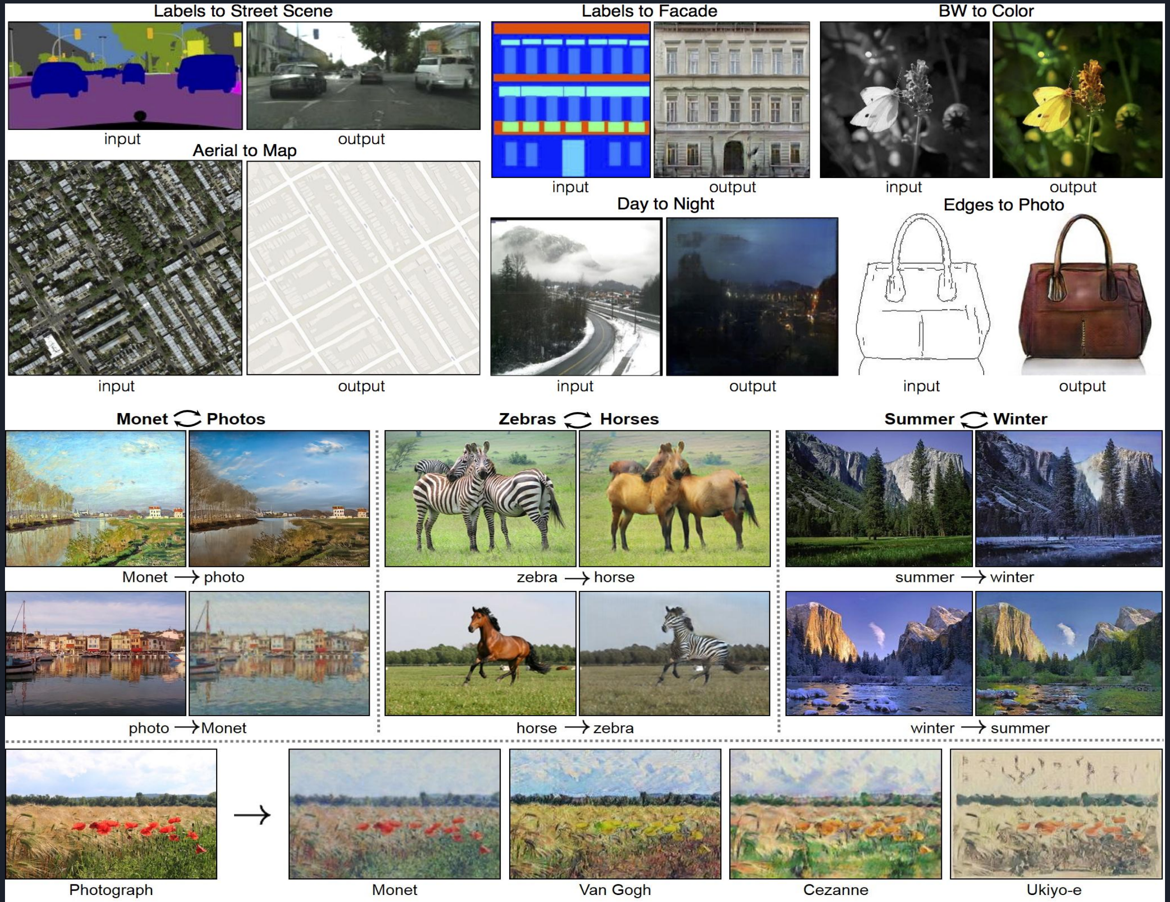


Modele generative

- Antrenarea modelelor generative poate determina reprezentări latente ce pot fi folosite ca feature-uri generale.

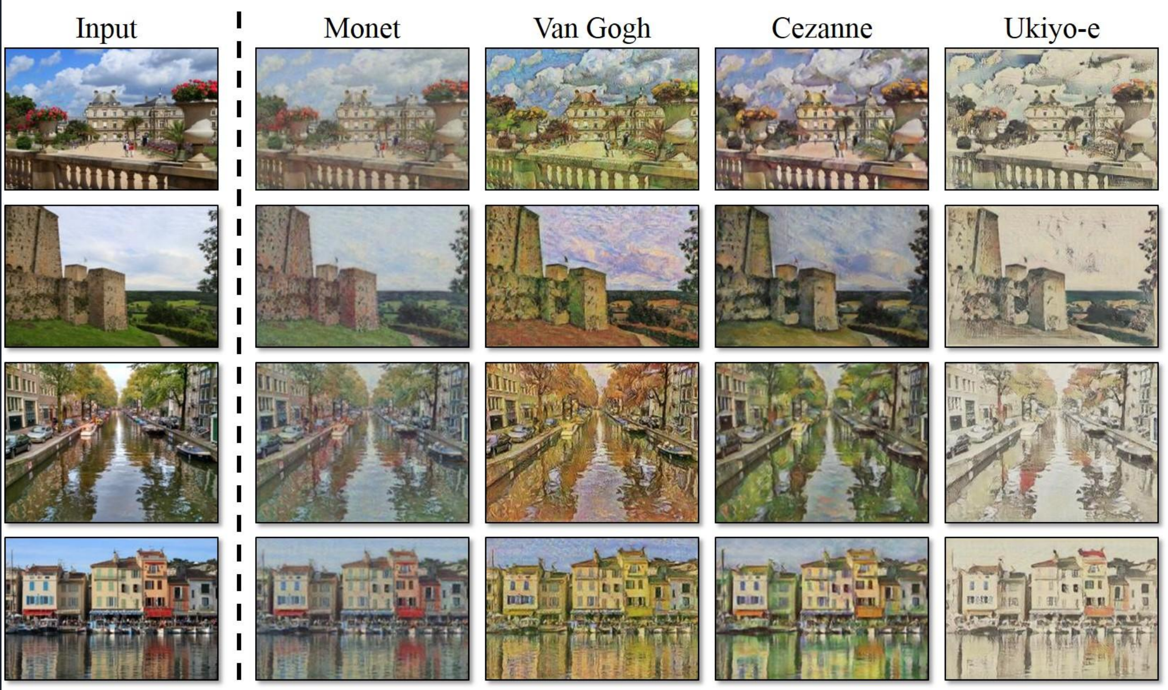
Modele generative

- Augmentare de date



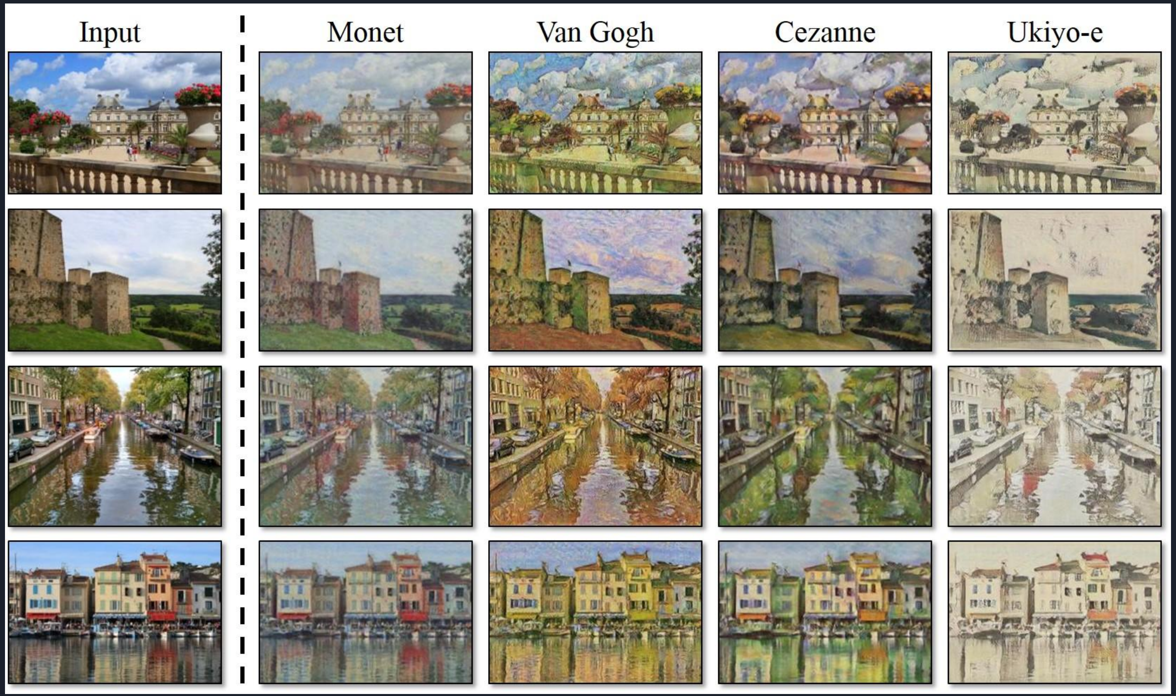
Modele generative

- Style transfer



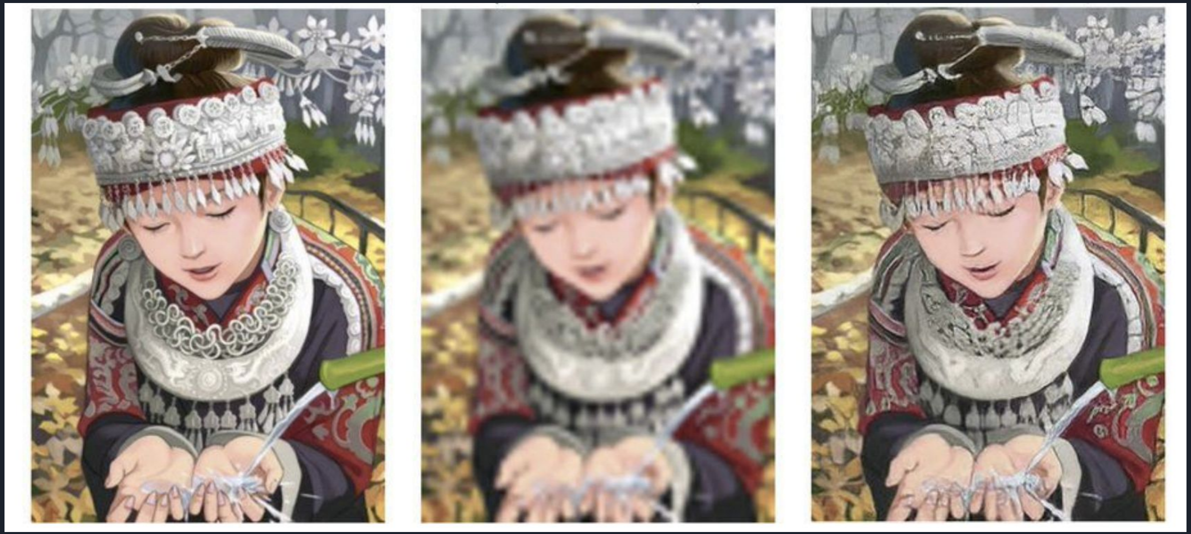
Modele generative

- Style transfer



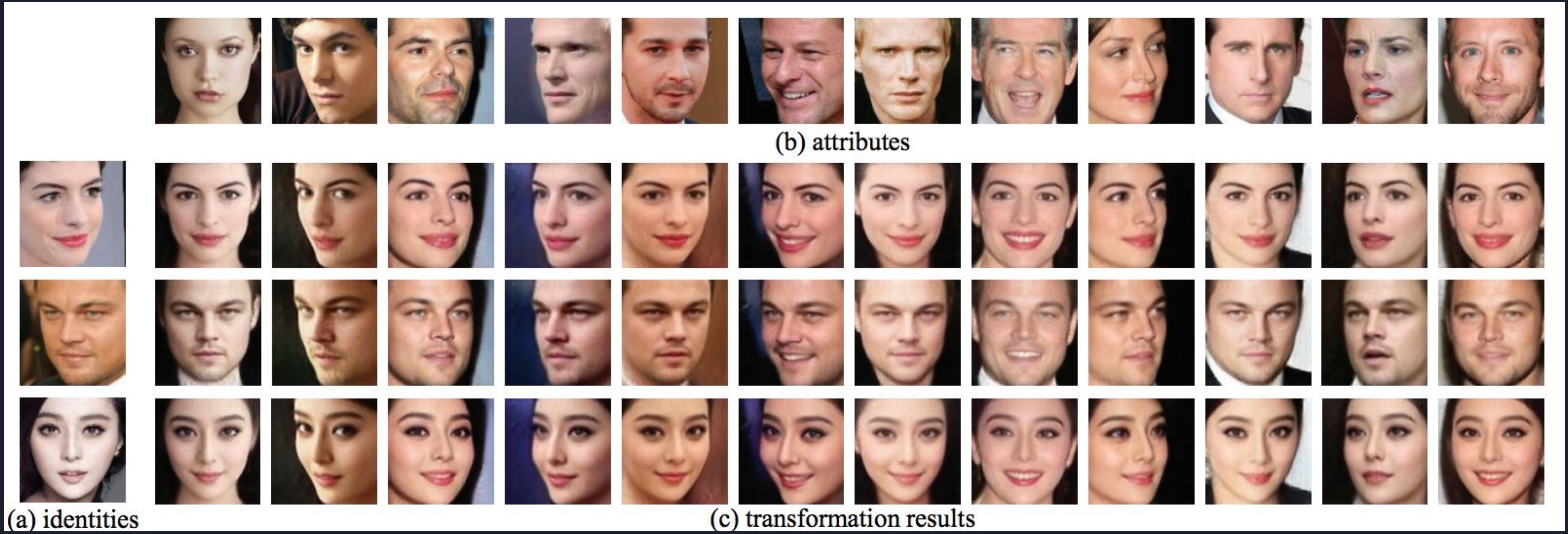
Modele generative

- Super-resolution



Modele generative

- Face attribute manipulation
- Face synthesis

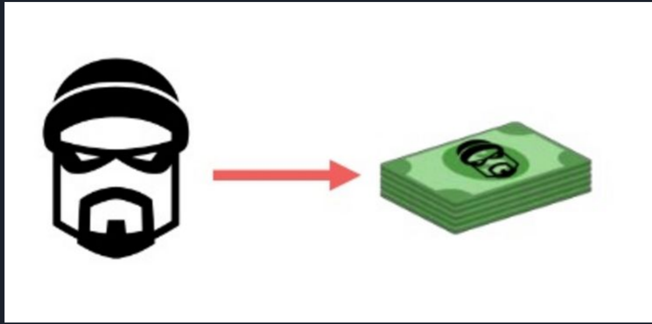


Abordari pentru modele generative adanci

- **Generative Adversarial Networks (GANs)**
  - GAN
  - Wasserstein GAN
- **Latent variable models**
  - Variational Autoencoders (VAEs)
- **Autoregressive models**
  - Deep NADE
  - PixelRNN
  - PixelCNN
  - WaveNet
  - Video Pixel Network

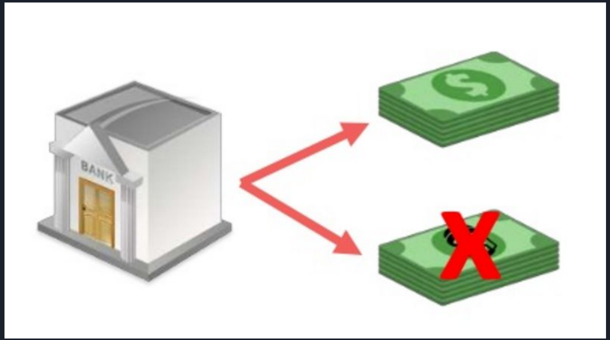
Generative Adversarial Networks (GANs) [1]

Generator



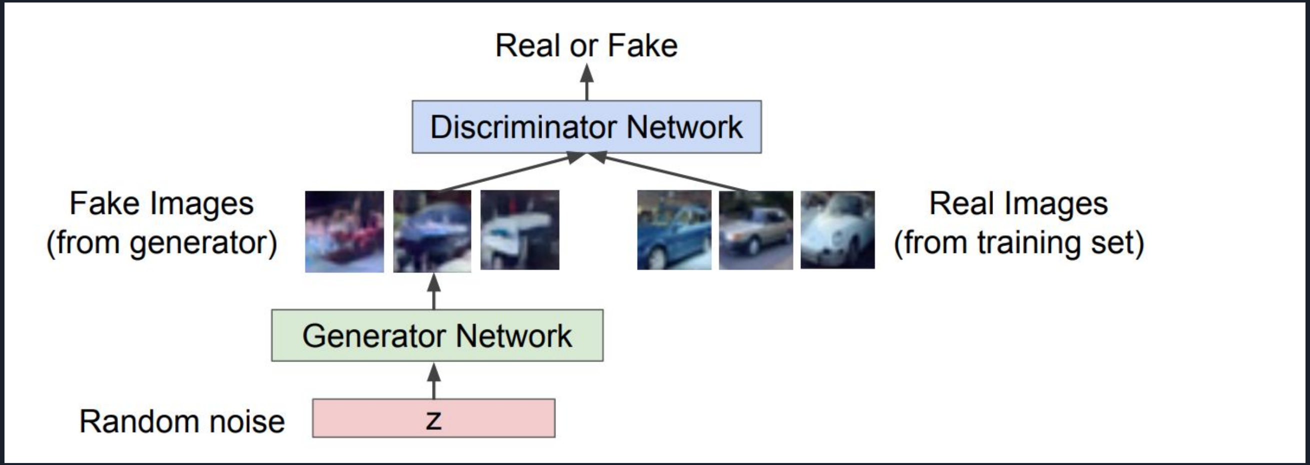
➢ Incearca sa genereze bancnote false cat mai asemanatoare cu cele reale

Discriminator



➢ Incearca sa distinga cat mai bine intre bancnote reale si cele false generate de catre Generator

Generative Adversarial Networks (GANs) [2]



(de obicei) generat prin samplarea unei distributii normale

Datele de antrenare (training data)

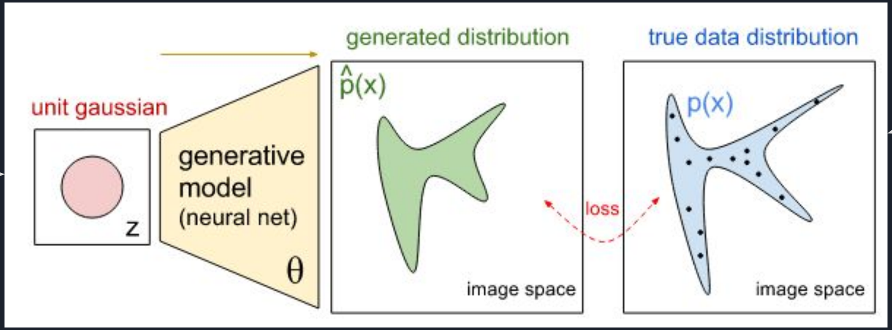
Generative Adversarial Networks (GANs) [3]

- **Generatorul** & **Discriminatorul** sunt representate de retele neuronale
- Cele doua retele sunt angrenate intr-un joc mini-max si incearca sa gaseasca un Nash Equilibrium
- **Generatorul** incearca sa genereze sample-uri cat mai reale astfel incat sa pacaleasca discriminatorul ca acestea sunt reale
- Discriminatorul incearca sa distinga cat mai bine intre sample-uri reale si date fake
- Antrenarea ar trebui sa se termine atunci cand
    - **Generatorul** reproduce exact distributia datelor reale
    - **Discriminatorul** alege random intre real/fake cu prob 0.5

Generative Adversarial Networks (GANs) [4]

Invatam o mapare (determinista)
intre z si imagini generate

Z este generat prin samplarea dintr-o gausiana multi-dimensionala



Avem un dataset de antrenare format din exemple: x1, x2, …. Xn samplate din distributia reala p(x)

- Goal-ul este sa modificam parametrii retelei ai sa producem o distributie care seamana cu cea reala
- Putem folosi KL divergence ca si functie obiectiv:

$$D_{\mathrm{KL}}(P\|Q) = -\sum_i P(i) \log \frac{Q(i)}{P(i)}$$

Antrenarea GAN-urilor [1]

- Jocul dintre G si D poate fi exprimat prin functia obiectiv minimax

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(1 - D(\tilde{\boldsymbol{x}}))]$$

Ouput discriminator pt date reale

Ouput discriminator pt date fake

  - *Pr* este distributia reala
  - *Pg* este distributia modelului din care generam sample-uri: $\tilde{\boldsymbol{x}} = G(\boldsymbol{z}), \quad \boldsymbol{z} \sim p(\boldsymbol{z})$
  - *p(z)* este o distributie simpla (normala)

- Alternand
  - Gradient ascent peste obiectivul discriminatorului $\max_{\theta_d}\left[\mathbb{E}_{x\sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))\right]$
  - Gradient descent peste obiectivul generatorului $\min_{\theta_g} \mathbb{E}_{z\sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$

discriminator se satureaza => vanishing gradients

Antrenarea GAN-urilor [1]

- Functia obiectiv minimax

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(1-D(\tilde{\boldsymbol{x}}))]$$

Ouput discriminator pt date reale

Ouput discriminator pt date fake

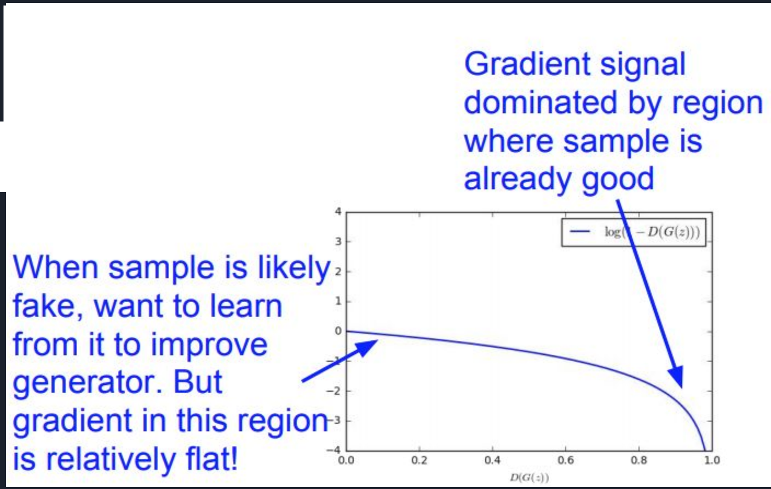
- Alternand
  - Gradient ascent peste obiectivul discriminatorului

$$\max_{\theta_d}\left[\mathbb{E}_{x\sim p_{data}}\log D_{\theta_d}(x) + \mathbb{E}_{z\sim p(z)}\log(1-D_{\theta_d}(G_{\theta_g}(z)))\right]$$

  - Gradient ascent peste obiectivul generatorului

$$\max_{\theta_g}\mathbb{E}_{z\sim p(z)}\log(D_{\theta_d}(G_{\theta_g}(z)))$$

In prezenta unui discriminator bun -> tot apar probleme in antrenarea generatorului



Antrenarea GAN-urilor

for number of training iterations **do**

for $\boxed{k \text{ steps}}$ **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:

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

end for

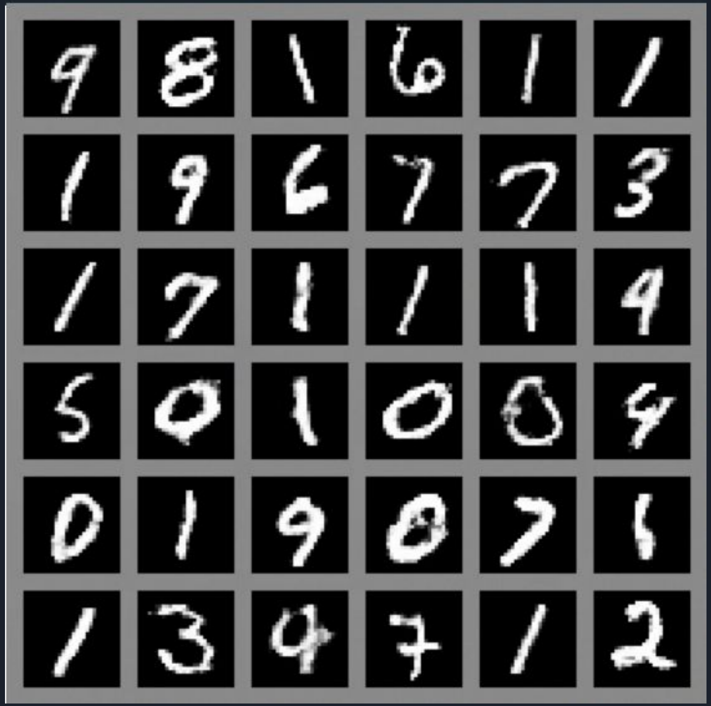
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by ascending its stochastic gradient (improved objective):

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Wasserstein GAN - mai stabila antrenarea

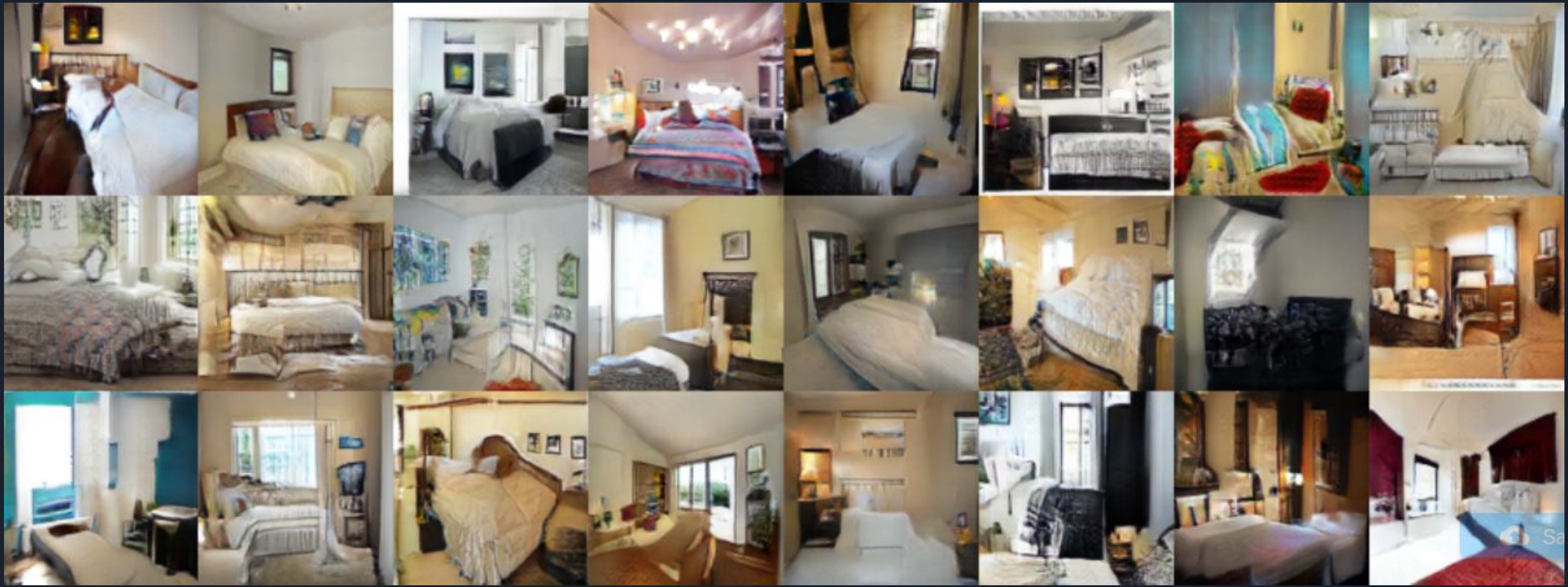
Sample-uri generate de GAN-uri



MNIST

CIFAR

Least-Squares GAN Xudong Mao, Qing Li†, Haoran Xie, Raymond Y.K. Lau and Zhen Wang, ArXiv, Feb. 2017

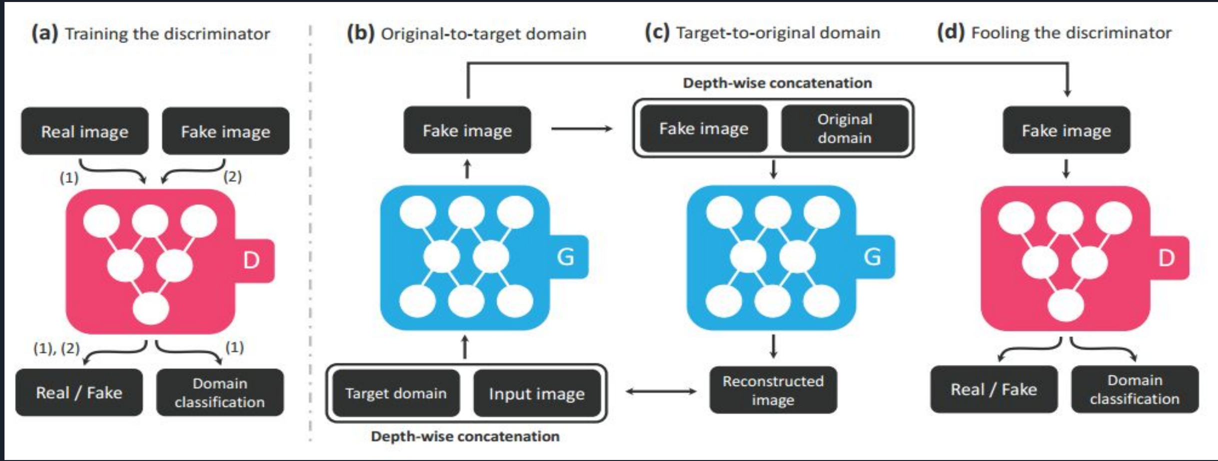


128x128 LSUN bedroom scenes

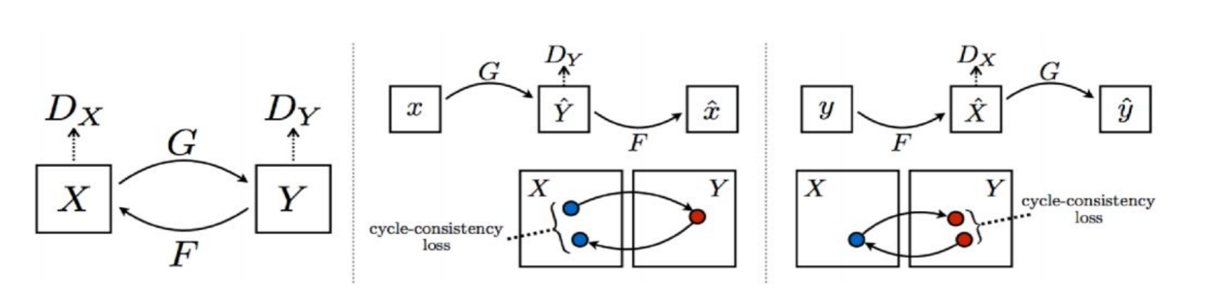
cycleGAN, StarGAN,

- Invata mapari 1-1 intre domenii folosind date ne-imperecheate (unpaired data)
- Folosesc cycle-consistency loss (reconstructie L1)

StarGAN



CycleGAN

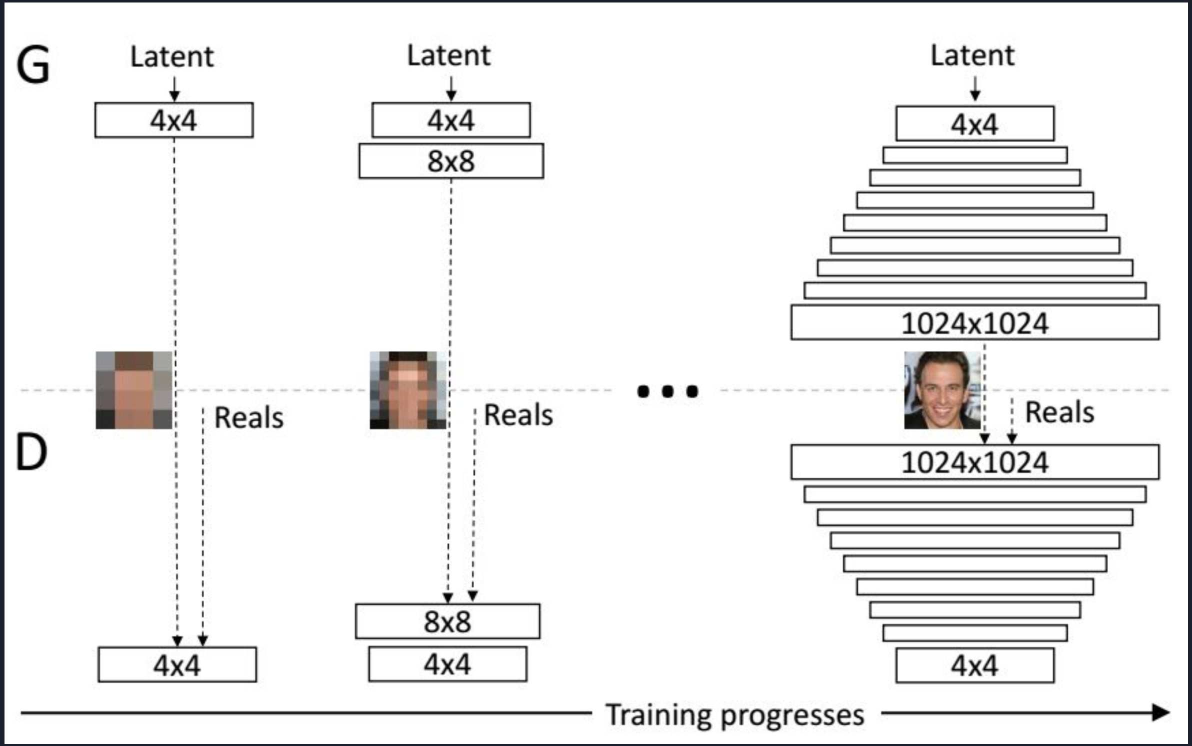


Progressive growing of GANs for improved quality, stability and variation (Kerras et al. from NVIDIA, 2017) [1]

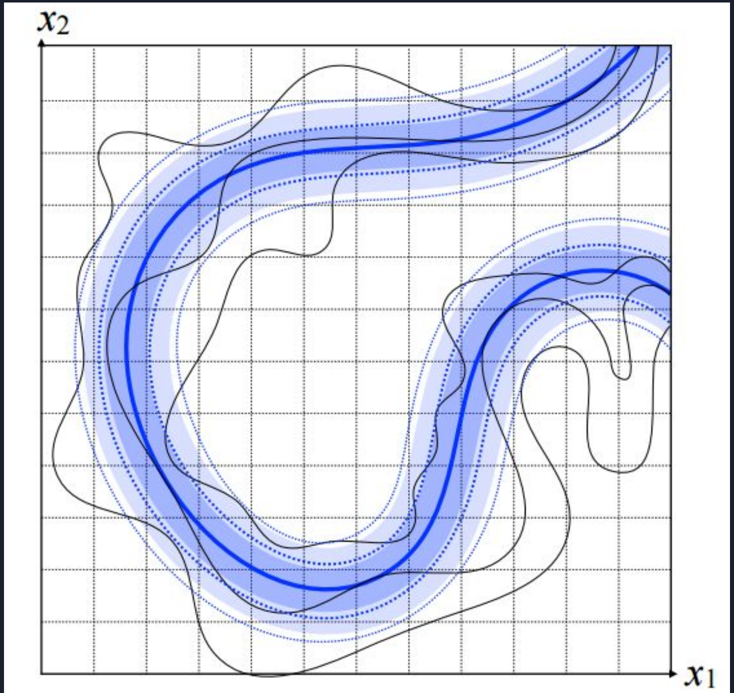
- Imbunatateste calitatea sample-urilor crescand progresiv marimea modelului in timpul antrenarii

- Sample-urile generate cu un model antrenat pe datasetul CelebA 1024x1024

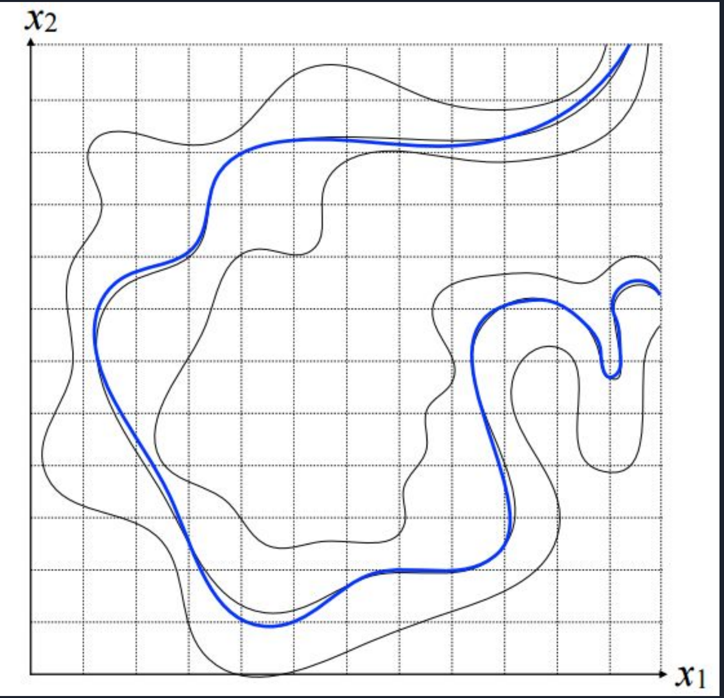
Progressive growing of GANs for improved quality, stability and variation (Kerras et al. from NVIDIA, 2017) [1]



Teoria GAN-urilor

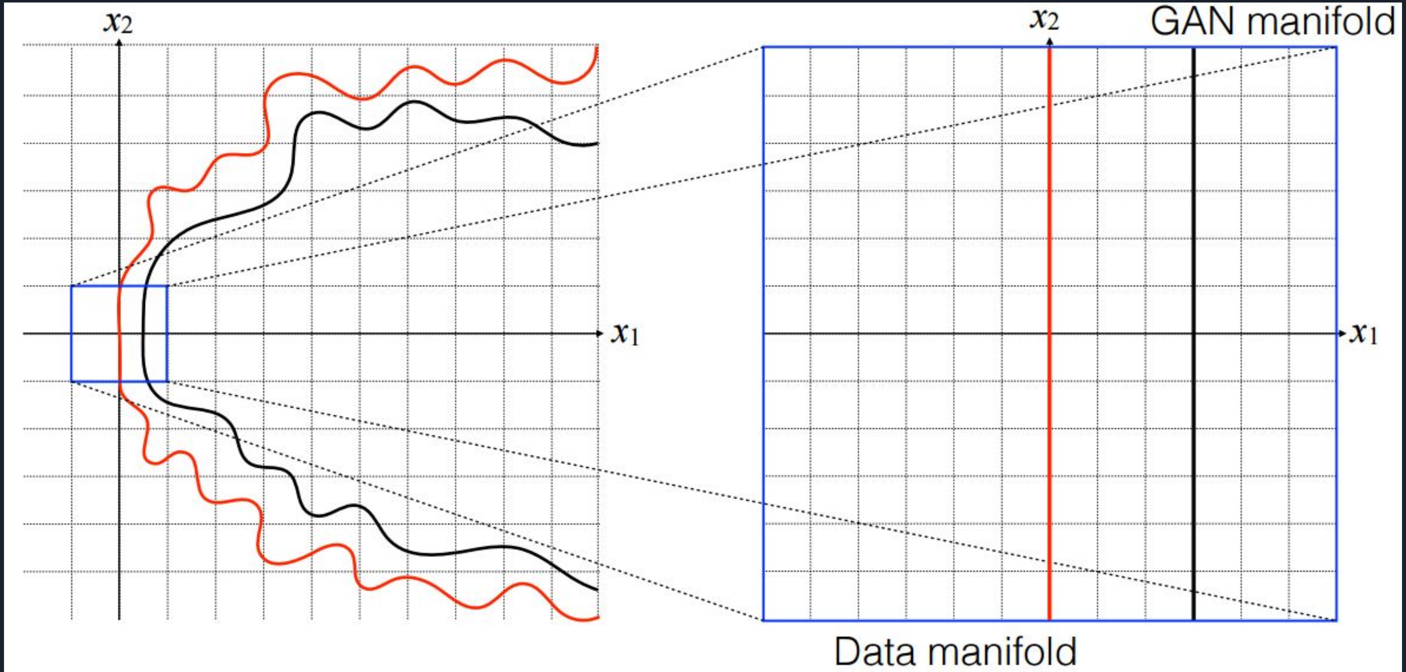


Metode traditionale - maximizarea probabilitatii de aparitie a datelor (maximum likelihood)



GAN

Antrenarea GAN-urilor: Distanta dintre manifolduri



Jensen-Shannon Divergence

$$\mathrm{JS}(\mathbb{P}_r \| \mathbb{P}_g) = \begin{cases} \log 2 & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0, \end{cases}$$

- GAN-urile optimizeaza divergenta Jensen-shannon (un middle ground intre doua functii de cost)

$$\mathrm{JS}(\mathbb{P}_r \| \mathbb{P}_g) = \mathrm{KL}\left(\mathbb{P}_r \middle\| \frac{\mathbb{P}_r + \mathbb{P}_g}{2}\right) + \mathrm{KL}\left(\mathbb{P}_g \middle\| \frac{\mathbb{P}_r + \mathbb{P}_g}{2}\right)$$

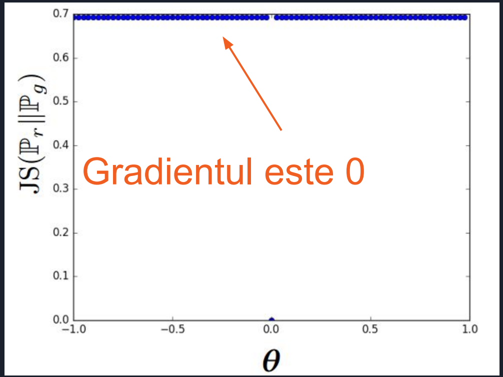
$$\mathrm{KL}(\mathbb{P}_r \| \mathbb{P}_g) = \int \log\left(\frac{p_r(x)}{p_g(x)}\right) p_r(x) d\mu(x)$$

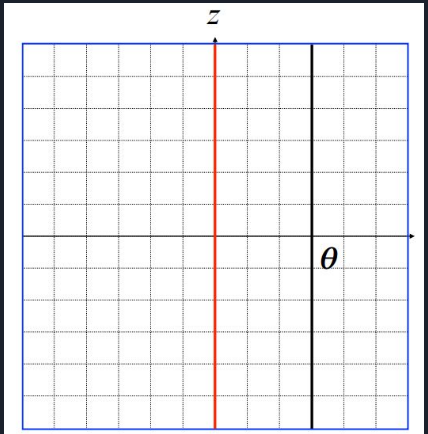
- Functia obiectiv a discriminatorului este:

$$L(D, g_\theta) = \mathbb{E}_{x \sim \mathbb{P}_r}[\log D(x)] + \mathbb{E}_{x \sim \mathbb{P}_g}[\log(1 - D(x))]$$

- Un discriminator optim are forma: $D^*(x) = \frac{P_r(x)}{P_r(x) + P_g(x)}$

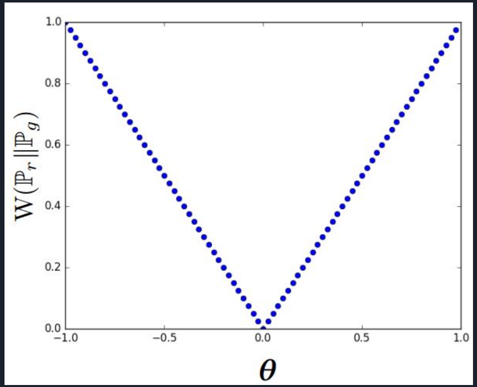
$$L(D^*, g_\theta) = 2JSD(\mathbb{P}_r \| \mathbb{P}_g) - 2\log 2$$

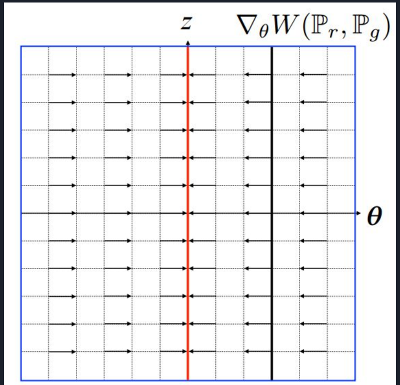




Distanta Earth-Movers

$$W(\mathbb{P}_r\|\mathbb{P}_g) = |\theta|$$





- Divergenta JS nu ofera semnal bun pentru antrenarea GAN-urilor
- Distanta Earth-Mover (Wasserstein-1)

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\|x - y\|\big]$$

  - Costum minim de a transporta masa din distributia *Pr* in distributia *Pg*
  - Este continua peste tot si diferentiabila aproape peste tot (sub anumite conditii)
- Kantorovich-Rubinstein duality

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)]$$

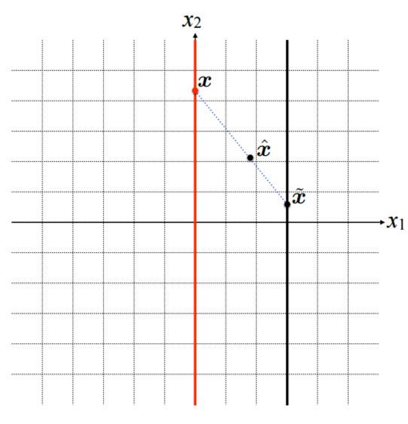
  - Supremum peste functii 1-Lipschitz f : X -> R
  - f => critic

WassersteinGAN

- Functia obiectiv

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \big[D(\boldsymbol{x})\big] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \big[D(\tilde{\boldsymbol{x}}))\big]$$

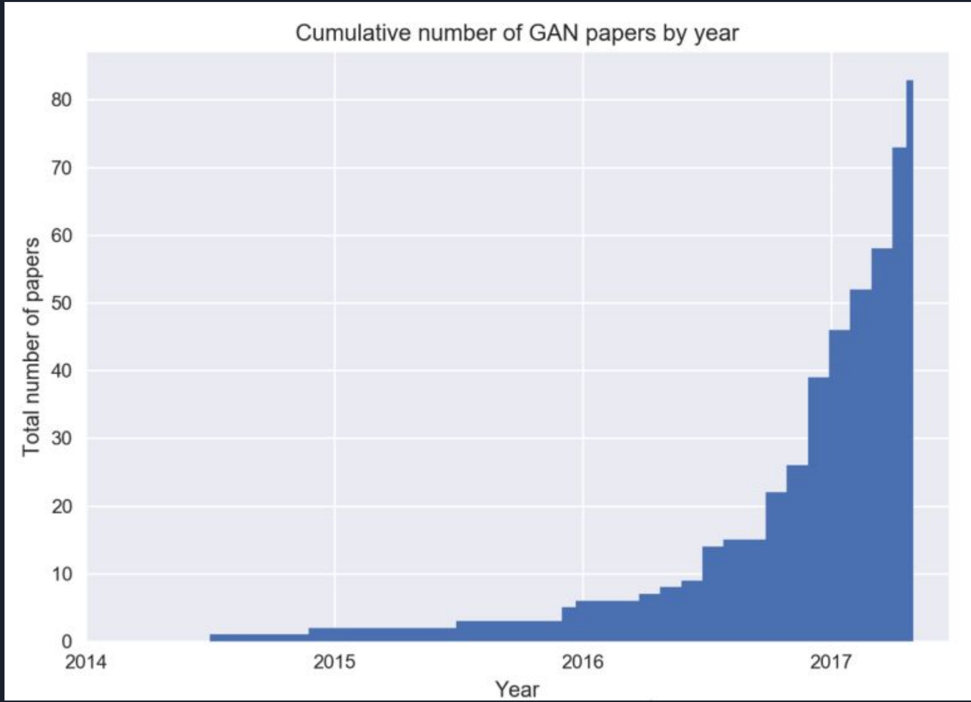
- D - 1-Lipschitz functions
- Cum putem sa punem constrangerea Lipschitz peste critic D?
  - Weight clipping [-c, c]
  - Gradient penalty



$$\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} \left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]$$

$$\epsilon \sim U[0,1], \boldsymbol{x} \sim \mathbb{P}_r, \tilde{\boldsymbol{x}} \sim \mathbb{P}_g$$
$$\hat{\boldsymbol{x}} = \epsilon \boldsymbol{x} + (1-\epsilon)\tilde{\boldsymbol{x}}$$

Explozia GAN-urilor



References

- https://drive.google.com/file/d/0ByUKRdiCDK7-bTgxTGoxYjQ4NW8/view?usp=drive_web - Generative Models 1
- https://drive.google.com/file/d/0B_wzP_JlVFcKQ21udGpTSkh0aVk/view?usp=drive_web - Generative Models 2
- https://arxiv.org/pdf/1701.04862.pdf - GAN theory
- https://arxiv.org/abs/1701.07875 - WassersteinGAN
- http://arxiv.org/pdf/1711.09020v1.pdf - StarGAN
- https://arxiv.org/abs/1703.10593 - CycleGAN
- https://arxiv.org/abs/1710.10196v3 - Progressive Growing of GANs for Improved Quality, Stability, and Variation
- https://arxiv.org/abs/1511.06434 - DCGAN
- https://arxiv.org/abs/1701.00160 - NIPS GAN tutorial