GAN Recitation

Hira Dhamyal, Simral Chaudhary, William Hu

Topics for today

- Introduction of GAN
- Types of GANs
- Earth Mover Distance
- Wasserstein Gan
- Example of GAN implementation

What are GAN

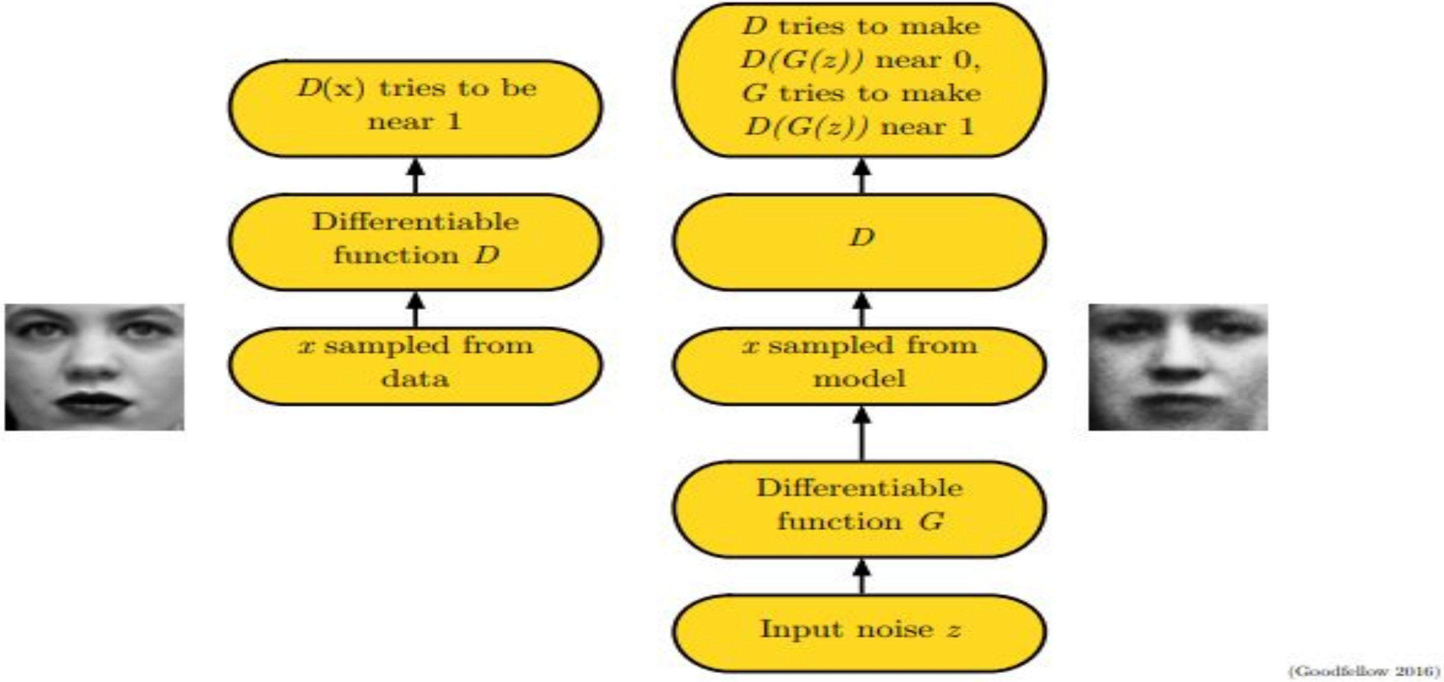
- GAN are are a way of generative modelling
- A machine that tries to understand what is the distribution of training samples given some observations from the distribution
- Observes training data and is able to generate more samples from the distribution learnt from the training data

Adversarial Training

- In Generative adversarial network, adversarial training refers to training a neural network to generate adversarial examples *by* training on adversarial examples
- Adversarial training includes two players
- In Generative adversarial network, both players are neural networks
- In worst case, the training sample is generated by one of the players

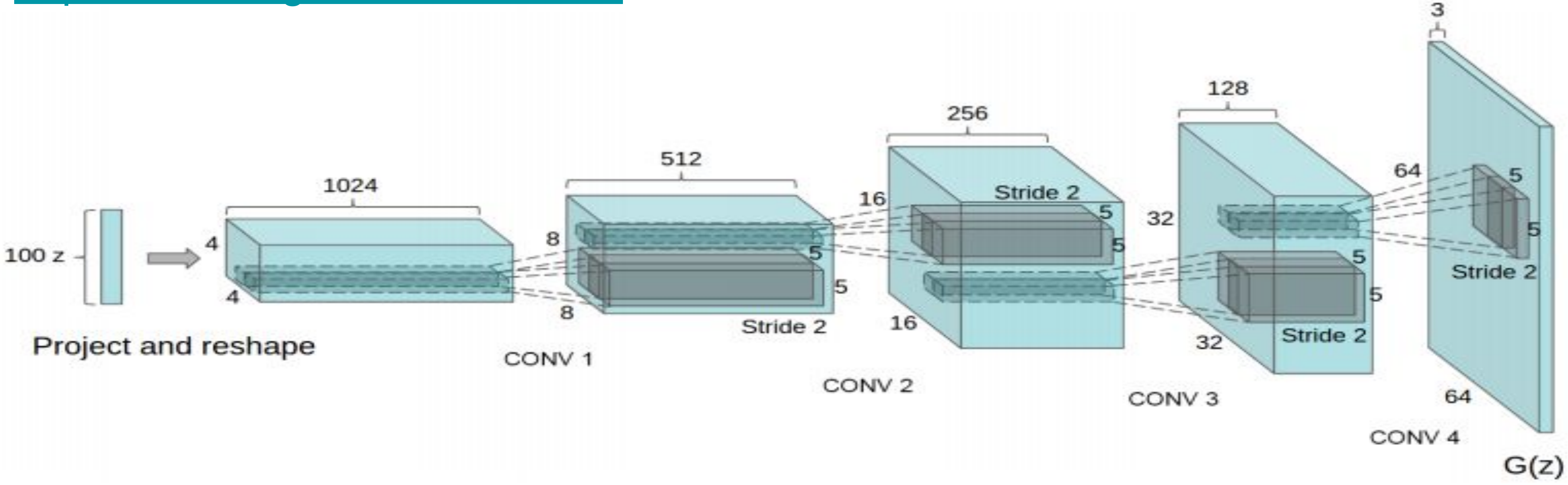
Adversarial Network Framework

(taken from Ian Goodfellow’s slides)



DCGan (ICLR 2016)

Paper can be found here;
https://arxiv.org/abs/1511.06434



DcGans Application

- Image generation
- Face generation
- Scene modeling

Pix2pix (2018)

Paper can be found here;
https://arxiv.org/pdf/1611.07004.pdf

- Image-to-Image Translation
- Conditions the output on an image sample x

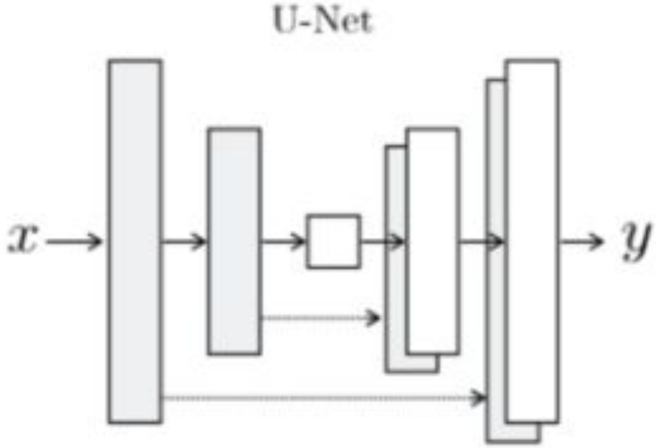
Nice article here:

https://towardsdatascience.com/pix2pix-gan-in-tensorflow-2-0-fe0ab475c713

https://phillipi.github.io/pix2pix/

Pix2pix

Generator
architecture



Pix2Pix Gan Applications

- Image coloring
- Image operations like background masking (http://www.k4ai.com/imageops/index.html)
- Video processing - removes the background in the video
- And many more ...

Cool Results

Fake faces generated using GANs

ThisPersonDoesNotExist.com.

AI uses a “512 dimensional vector” to generate a new facial image

Paper: https://arxiv.org/pdf/1812.04948.pdf

Probability theory behind Generative models

- Unknown distribution P_r (r for real)
- Known distribution P_θ (θ parameterised)
- Two approaches:
  - Optimise P_θ to estimate P_r
  - Learn a function $g_\theta(Z)$ which transforms Z into P_θ

Approach 1: Optimise P_θ to estimate P_r

- How?
  - Maximum Likelihood Estimation (MLE) $$\max_{\theta \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} \log P_\theta(x^{(i)})$$
- Equivalent to minimizing the KL-divergence $KL(P_r \,||\, P_\theta)$
- Issue: Exploding of KL-divergence for zero values of P_θ
  - Fix: Add random noise to P_θ
  - Why go through all the trouble?

Approach 2: Learn a function $g_\theta(Z)$ which transforms Z into P_θ

- Z is a known distribution s.a Uniform or Gaussian distribution
- We learn a generator function which will transform this Z into P_θ
- How to train g_θ (and eventually P_θ)?
  - Minimize distance between g_θ and P_r
- Distance metrics
- Loss function: $d(P_r, P_\theta)$

Distance Metrics

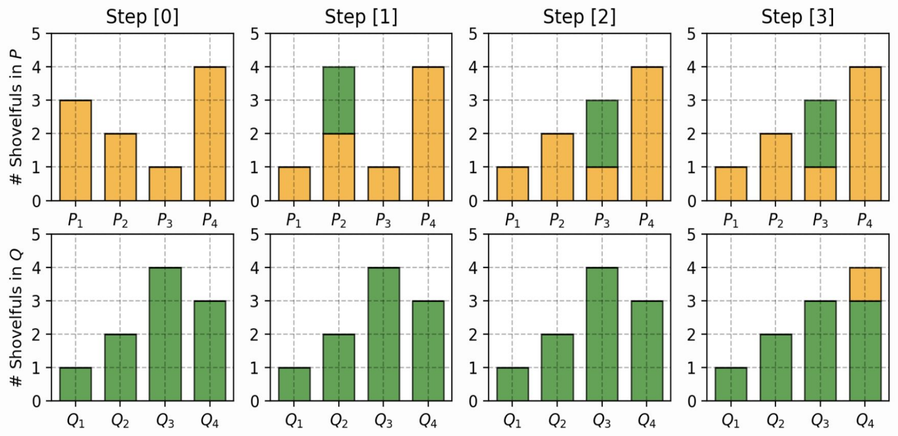
- Total Variation (TV) distance $$\delta(P_r, P_g) = \sup_A |P_r(A) - P_g(A)|$$
- Kullback-Leibler (KL) divergence $$KL(P_r \| P_g) = \int_x \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x)\, dx$$
- Jenson-Shannon (JS) divergence $$JS(P_r, P_g) = \frac{1}{2} KL(P_r \| P_m) + \frac{1}{2} KL(P_g \| P_m)$$
- Earth Mover (EM) or Wasserstein distance:

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma} \left[\, \|x - y\| \,\right]$$

Let $\Pi(P_r, P_g)$ be the set of all joint distributions γ whose marginal distributions are P_r and P_g

Earth Mover distance (Wasserstein distance)

- **Earth Mover's distance** : the minimum energy cost of moving and transforming a pile of dirt in the shape of one probability distribution to the shape of the other distribution.
- The cost is quantified by: the amount of dirt moved * the moving distance.
- Example 1:
  - P, Q : 4 piles of dirt made up of 10 shovelfuls of dirt present in each.
  - The numbers of shovelfuls in each dirt pile are assigned as follows:
    - $P_1 = 3, P_2 = 2, P_3 = 1, P_4 = 4$
    - $Q_1 = 1, Q_2 = 2, Q_3 = 4, Q_4 = 3$
    - $W = \sum |\delta i| = 5$



Source: Link

- Example 2:
  - Π(Pr, Pg) is the set of all possible joint probability distributions between Pr and Pg
  - $\gamma \in \Pi$(Pr, Pg) : one dirt transport plan
  - $$\sum_{x,y} \gamma(x, y)\|x - y\| = \mathbb{E}_{x,y\sim\gamma}\|x - y\|$$

Wasserstein GAN

- Use Wasserstein distance as GAN loss function
- The “discriminator” model does not play as a direct critic but a helper for estimating the Wasserstein metric between real and generated data distribution.
- Still not perfect :(WGAN still suffers from unstable training and other convergence issues.