



GANs (Generative Adversarial Networks)

By Amit, Parth and Hariharan



Agenda

- Generative models
- Revisiting GANs
- WGAN
- WGAN-Gradient penalty (WGANGP)
 - Code walk through GANS, WGAN, WGANGP
- Cycle GAN
 - Code walk through Cycle GAN
- STAR GAN
 - Code walk through STAR GAN



Generative Models

Basic idea is to learn the underlying distribution of the data and generate more samples for the distribution.

Some examples of generative models

- Probabilistic Graphical Models
- Bayesian Networks
- Variational Autoencoder
- Generative Adversarial Networks



Generative Models

- Unknown distribution P_r (r for real)
- Known distribution P_θ
- 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

- Maximum Likelihood Estimation (MLE) : $\max_{\theta \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^m \log P_\theta(x^{(i)})$
 - This is same as minimizing the KL divergence
- Kullback-Leibler (KL) divergence: $KL(P||Q) = \int_x \log\left(\frac{P(x)}{Q(x)}\right)P(x)dx$
- Issue: Exploding of KL-divergence for zero values of P_θ
 - Add random noise to P_θ



Approach 2: Learn a function $g_{\theta}(z)$

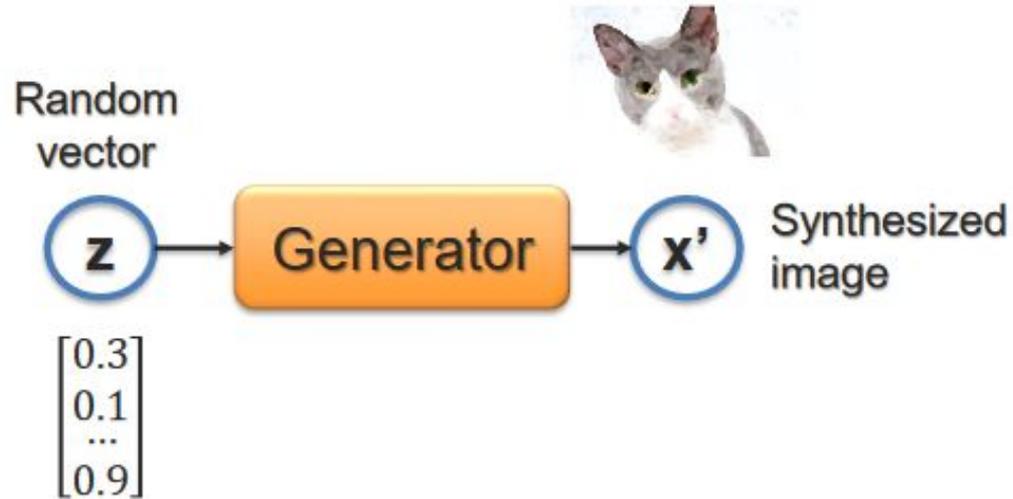
- We learn a function $g_{\theta}(z)$ that transforms z into P_{θ}
 - Z is a known distribution like Uniform or Gaussian
- We train g_{θ} by minimizing the distance between g_{θ} and P_r
- Any of the distance metrics like KL divergence, JS divergence or Earth Mover (EM) distance can be used.



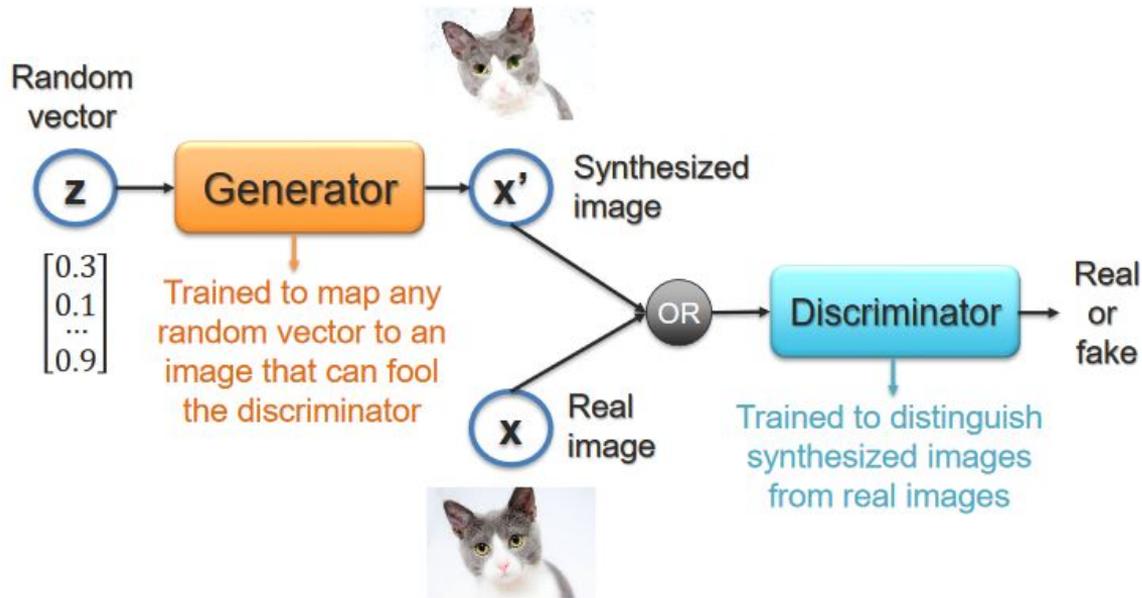
Revisiting GANs

- GANs are generative models which try to understand underlying distribution to generate more sample.
- GANs typically have 2 networks trained in an adversarial fashion.
 - Generator
 - Discriminator

Revisiting GANs- Generative Network



Revisiting GANs- Generator + Discriminator

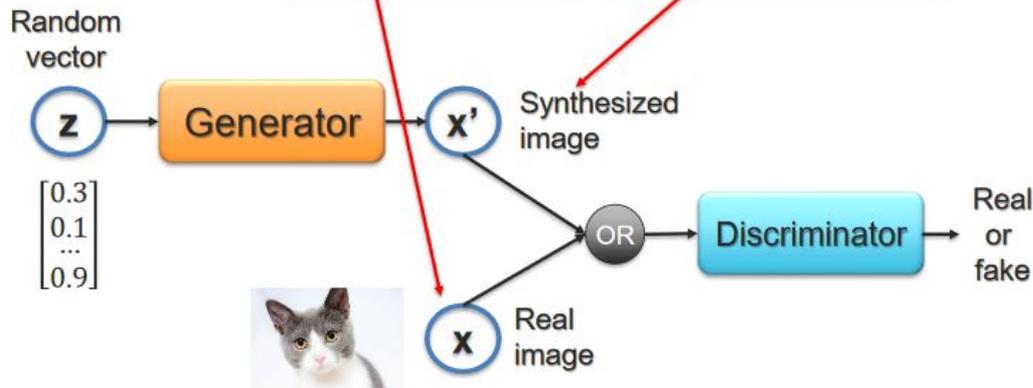


Revisiting GANs - training

$$\max_{\mathcal{D}} \min_{\mathcal{G}} V(\mathcal{G}, \mathcal{D})$$

How do we optimize
this objective function?

$$V(\mathcal{G}, \mathcal{D}) = \mathbb{E}_{p_{data}(\mathbf{x})} \log \mathcal{D}(\mathbf{x}) + \mathbb{E}_{p_g(\mathbf{x})} \log(1 - \mathcal{D}(\mathbf{x}))$$

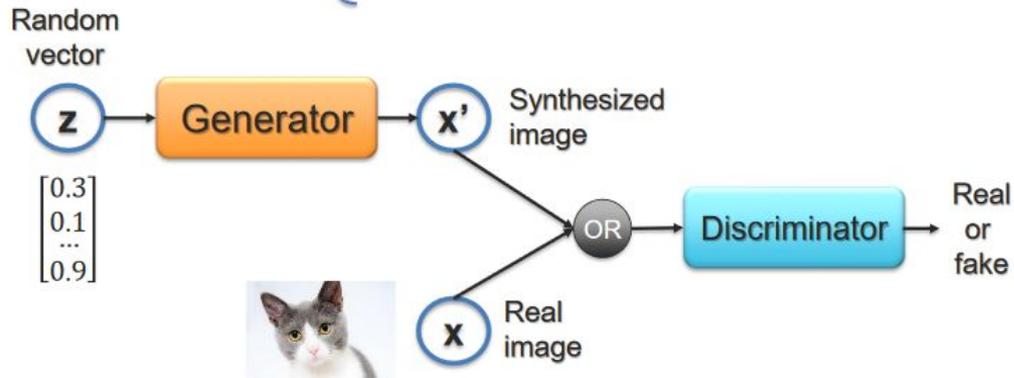


Revisiting GANs - training

$$\max_{\mathcal{D}} \min_{\mathcal{G}} V(\mathcal{G}, \mathcal{D})$$

Optimization:

- 1 Fix generator, and update discriminator
- 2 Fix discriminator, and update generator

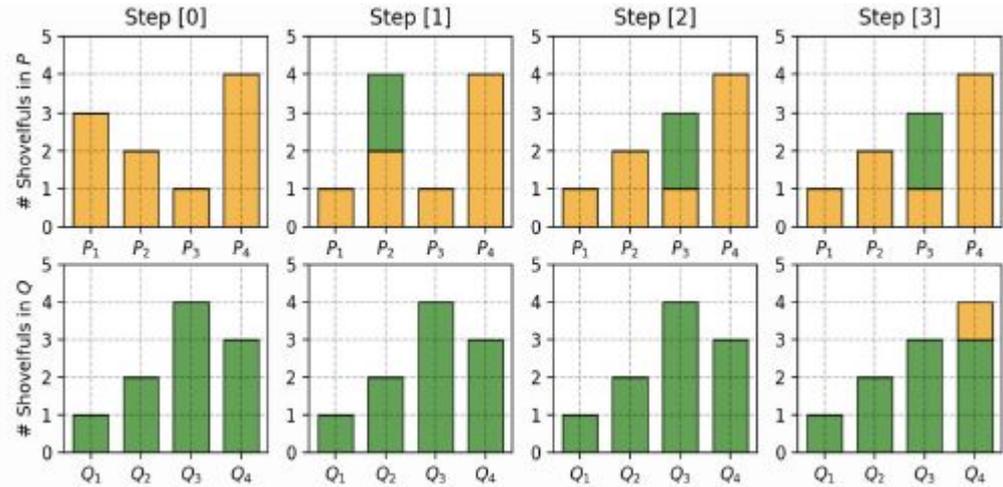


WGANs-Earth Mover Distance

Wasserstein distance: the minimum energy cost of moving and transforming a pile of dirt in the shape of one probability distribution to the the shape of other distribution.

P and Q: 4 piles of dirt made up of 10 shovelfuls of dirt present.

- ❑ $P_1 = 3, P_2 = 2, P_3 = 1, P_4 = 4$
- ❑ $Q_1 = 1, Q_2 = 2, Q_3 = 4, Q_4 = 3$
- ❑ $W = 5$





WGANs-Objective function

- We train GANs using this wasserstein distance.
- Discriminative is no more a direct critic. It is trained to estimate the wasserstein distance between real and generated data.

$$L_D = E_X D(X) - E_Z D(G(Z))$$

- Lipschitz is clipped to 1 i.e. $|f(x) - f(y)| / (x - y) \leq 1$
 - This bound on discriminator is not good, instead we clip the gradients.



WGAN-Gradient Penalty

- Bound on discriminator is not great and leads to poor discriminator.
- We can add the gradient penalty in the loss function making sure that the lipschitz is almost 1 everywhere.

$$L_D = E_X D(X) - E_Z D(G(Z)) + \lambda E_{X'} (\|\nabla D(X')\|_2 - 1)^2$$

- *We do not constraint the gradients everywhere.*
 - *We penalize where there is linear interpolation between real and fake data.*



Code Walkthrough

GANs, WGAN-GP



Image translation

- Image-to-image translation involves generating a new synthetic version of a given image.
- Example: Changing a summer landscape --> winter landscape, blonde --> black hair, image --> painting.
- Data for such image translation is very limited or sometimes difficult to generate.
- 2 variants of GANs are used for this specific task.
 - Cycle GAN
 - STAR GAN

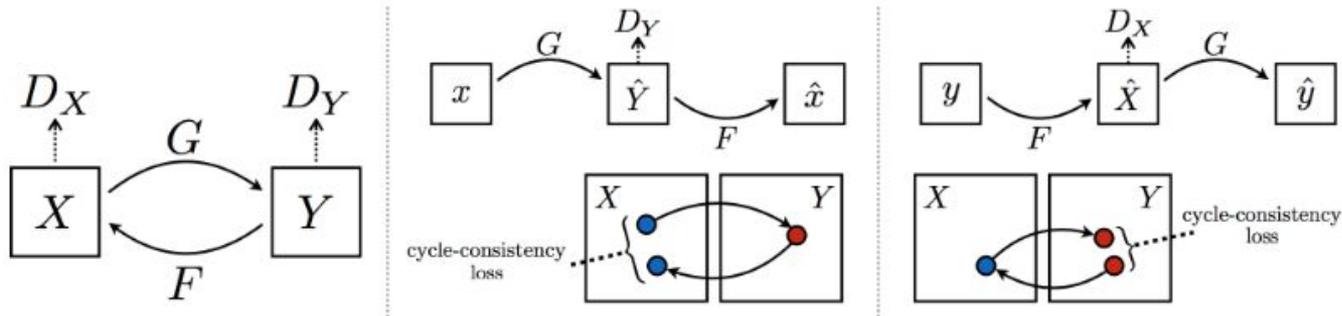


Cycle GANs

- Instead of a single Generator-Discriminator we have two Generators and discriminators.
 - One generator takes images from the first domain and outputs images from the second domain.
 - Discriminator models are used to determine how plausible generated images are and update the generator accordingly.
- The overall loss function for the cycle GAN is given below apart from the standard objective we have an added cycle-consistency loss.

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = & \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ & + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ & + \lambda \mathcal{L}_{\text{cyc}}(G, F),\end{aligned}$$

Cycle GAN



Cycle-consistency loss:
$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbf{E}_{x \sim p_{\text{data}}(x)} [\|F(G(x)) - x\|_1] + \mathbf{E}_{y \sim p_{\text{data}}(y)} [\|G(F(y)) - y\|_1].$$

Application: Style Transfer



Application: Object Transfiguration



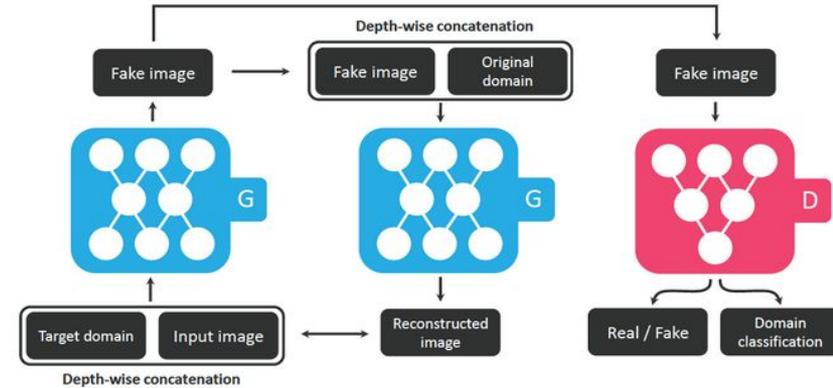


Star GAN (Unified GAN for Multi-Domain I2I translation)

- Star GAN helps us to generate images in target domain given an input and target domain.
 - Image of a man and target domain is gender.
 - Image of a person and target domain is age.
- We train the generator-discriminator in adversarial fashion with an added auxiliary classifier.
- Along with normal adversarial loss this loss is added while training the generator and discriminator.

Star GAN - Generator

- Generator have 3 objectives:
 - Tries to generate realistic images
 - The weights of generator are adjusted so that the generated images are classified as target domain by the discriminator.
 - Construct original image from the fake image given the original label domain label.



Objective
function:

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls} \mathcal{L}_{cls}^f + \lambda_{rec} \mathcal{L}_{rec}$$

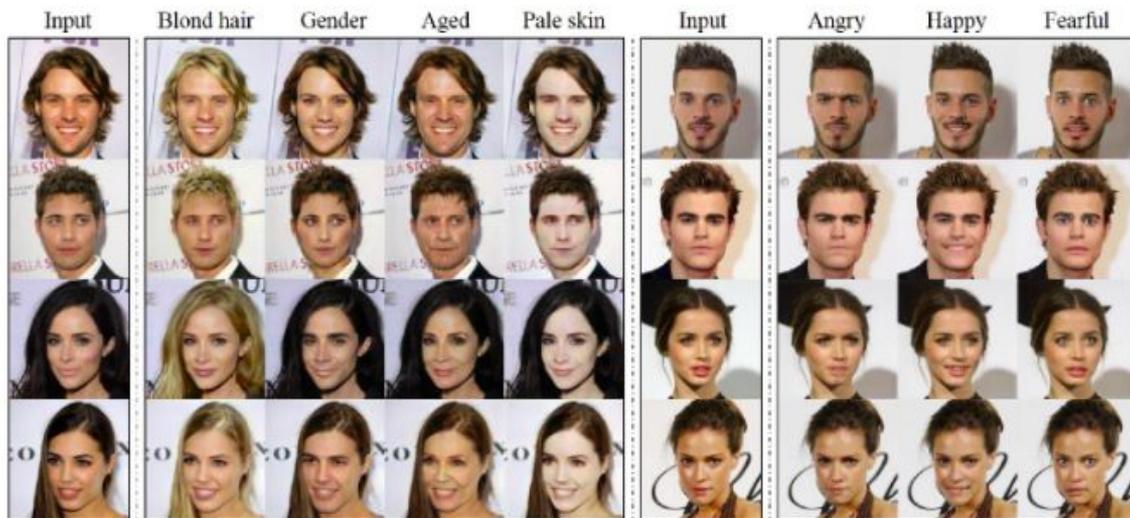


Star GAN - Discriminator

- Discriminator has 2 objectives:
 - Whether the image is fake or real
 - What is the domain in which the image belongs.
- If the generator is able to generate fool the discriminator then discriminator would predict the target domain and we stop training.

Objective function: $\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls} \mathcal{L}_{cls}^r$

Applications





Thank You!



Thank You!

Slow and steady wins the race is a lie, so pace up: Amit



Code Walkthrough

Cycle GAN and STAR GAN



References

- <https://arxiv.org/abs/1701.07875> (Wasserstein GAN)
- <https://arxiv.org/abs/1703.10593> (Cycle GAN)
- <https://arxiv.org/abs/1711.09020> (Star GAN)
- <https://machinelearningmastery.com/what-is-cyclegan/>
- <https://towardsdatascience.com/stargan-image-to-image-translation-44d4230fbb48>
- Lecture notes of 11-777