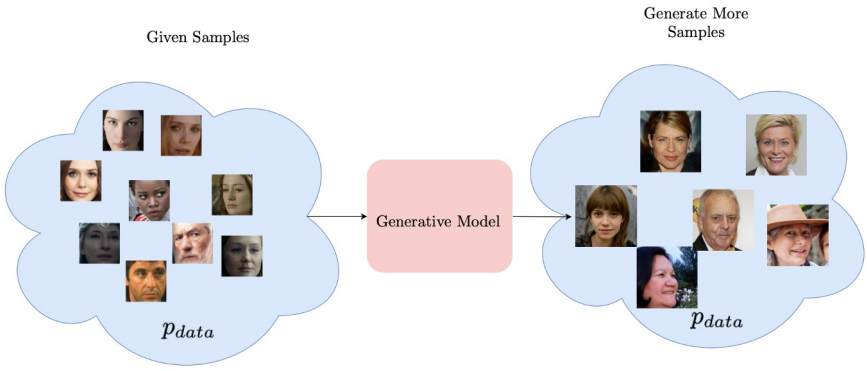
Generative Models

1

1

Generative Modelling

How does one sample from a distribution given only its samples?

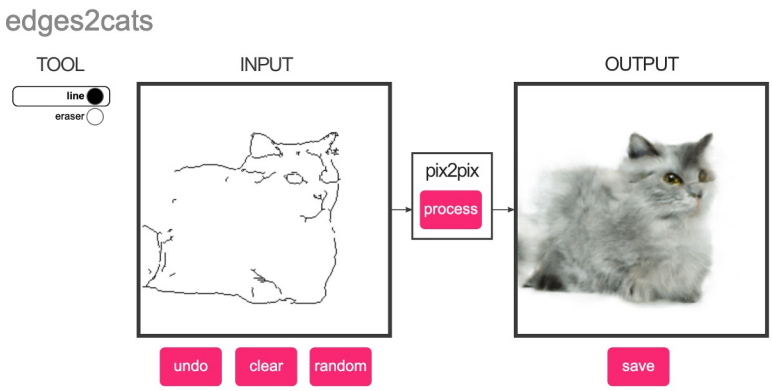


2

2

Generative Modelling: Why?

We can do some cool stuff! Like generate images from line drawings



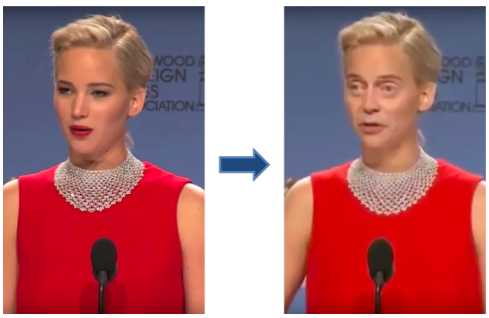
Demo: https://affinelayer.com/pixsrv/

3

3

Generative Modelling: Why?

Change faces!



4

4

Generative Modelling: Why?

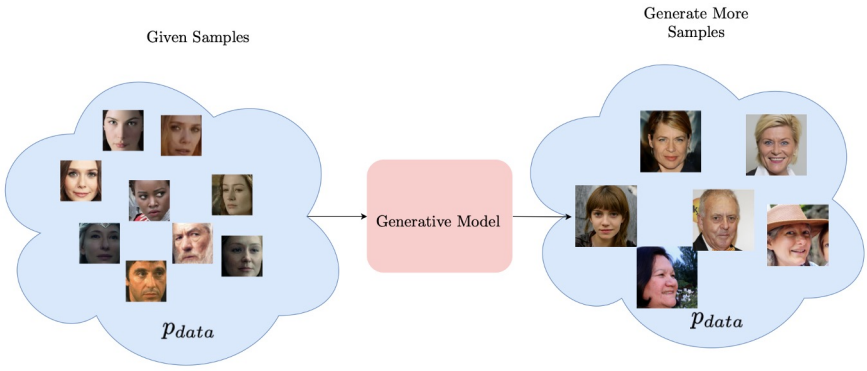
Or edit faces!

5

5

Generative Modelling

Right, so how does one sample from a distribution given only its samples?

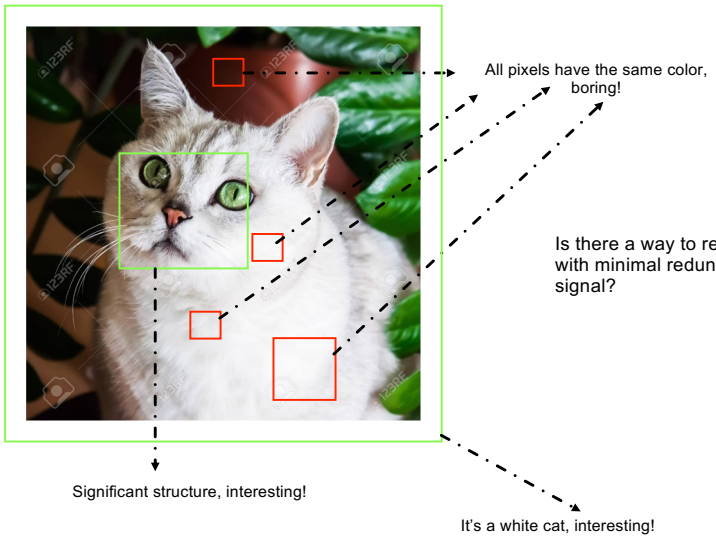


6

6

Generative Modeling: A Closer look at images

They're redundant, like very redundant



7

7

Generative Modeling: A Closer look at images

More realistically, is there a way to represent this image with minimal redundancy and a "good amount" of signal?

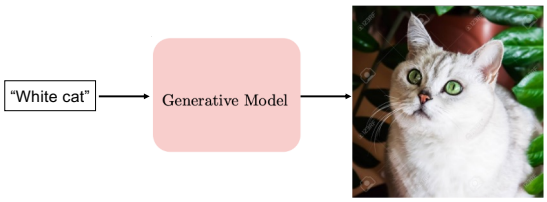
Yes! we do it all the time! Language!

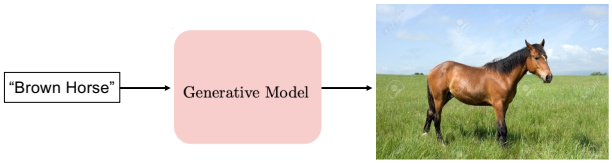
8

8

Generative Modeling: An ideal generative model

Say what you want the model to generate, and it generates it!





We can generate whatever we want!

9

9

Generative Modeling – From CSE527, 2020

"Unfortunately, life ain't that good and training such a model is hard. Especially because, text is quite a lossy representation" – Me, 2020

"White cat"

Eye Color? Head Shape? Number of whiskers?
Background?……

10

10

Generative Modeling – From CSE527, 2020

"Unfortunately, life ain't that good and training such a model is hard. Especially because, text is quite a lossy representation" – Me, 2020

11

11

Generative Modeling – From CSE527, 2020

"Unfortunately, life ain't that good and training such a model is hard. Especially because, text is quite a lossy representation" – Me, 2020

Right……

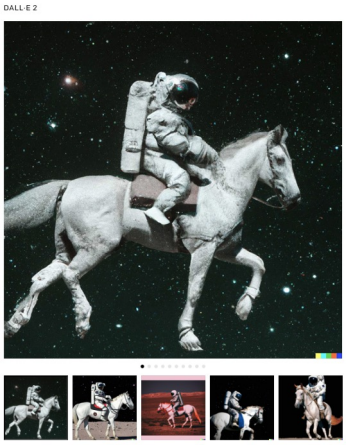
"A photorealistic image of a white cat on a garden ledge"

12

12

Generative Modeling

So, how do we get to DALL-E or Imagen? Lets start out journey…



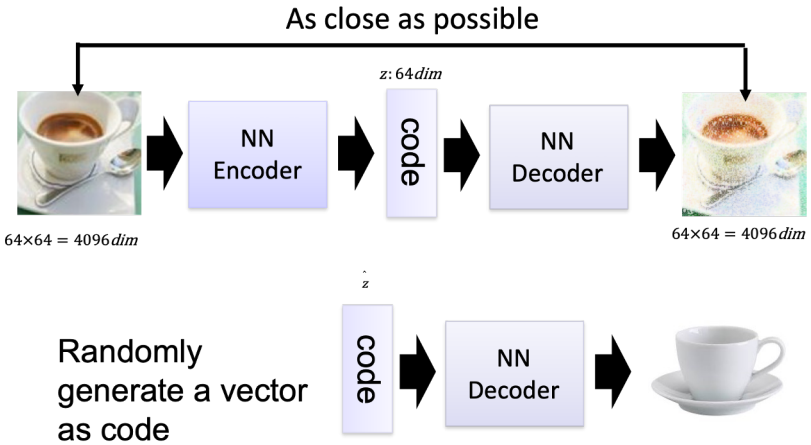
"An astronaut riding a horse in a photorealistic style"

13

13

Generative Modeling: Autoencoders

Learn a representation by compressing an image as much as possible and then reconstructing it.

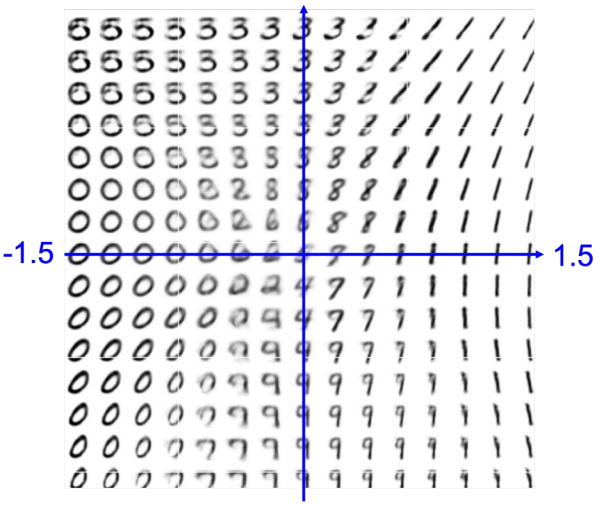


14

14

Generative Modeling: Autoencoders

But where do we sample from? A random vector may not produce a realistic image

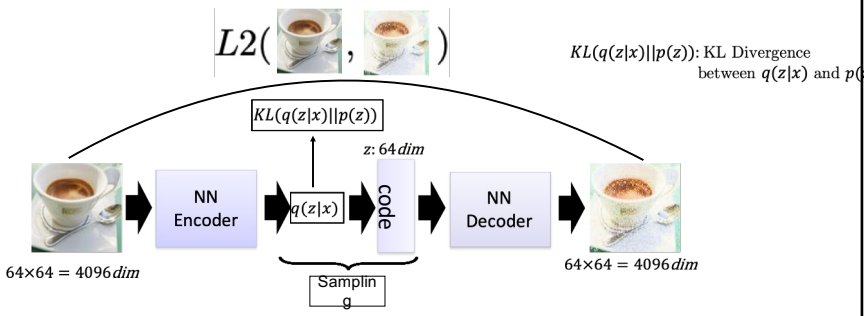


15

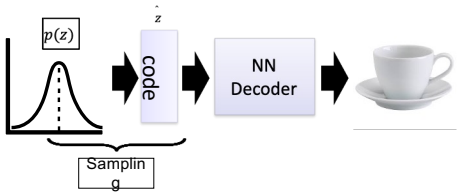
15

Generative Modeling: Variational Autoencoders

Learn a representation by compressing an image as much as possible and then reconstructing it, and push the distribution of the latent representation (i.e z) to a known prior $p(z) \sim N(0, I)$



Images can now be generated from a known prior distribution $p(z)$!



16

16

Generative Modeling: A diversion into divergences

A divergence allows one to measure the "distance" between two distributions $p(x)$ and $q(x)$. It is not a true distance measure because it may not be symmetric nor satisfy the triangle inequality. They satisfy the following properties:

- $D(p(x)||q(x)) \geq 0 \; \forall \; p(x), q(x)$
- $D(p(x)||q(x)) = 0 \textit{ if and only if } p(x) = q(x)$

Therefore, even if not a true distance, a divergence can allow us to measure how similar or dissimilar two distributions $p(x)$ and $q(x)$ are!

17

17

Generative Modeling: The KL and Jenson-Shanon Divergence

A divergence we've already seen is the KL-Divergence or Kullback-Leibler divergence which is defined as follows:

$$KL(p(x)||q(x)) = \int_{-\infty}^{\infty} p(x) log\left(\frac{p(x)}{q(x)}\right) dx$$

Notice $KL(p(x)||q(x)) \neq KL(q(x)||p(x))$!

This animation should give you some intuition about how the KL-Divergence works!

The Jenson-Shanon Divergence (JSD) is a symmetric version of the KL-Divergence and is defined as follows

$$JSD(p(x)||q(x)) = \frac{1}{2} KL\left(p(x)||\frac{p(x)+q(x)}{2}\right) + \frac{1}{2} KL\left(q(x)||\frac{p(x)+q(x)}{2}\right)$$

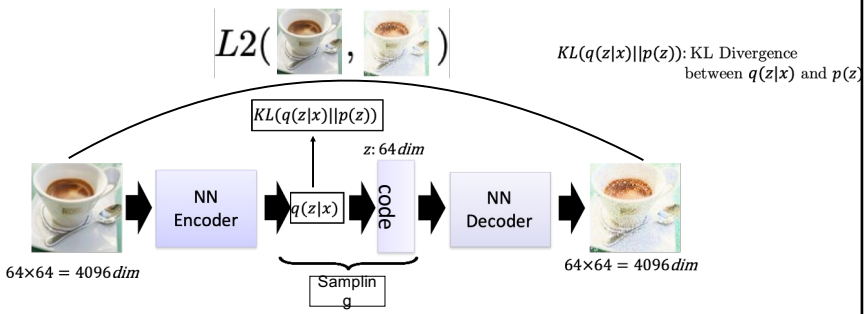
We'll be running into JSD when discussing GANs 🙂

18

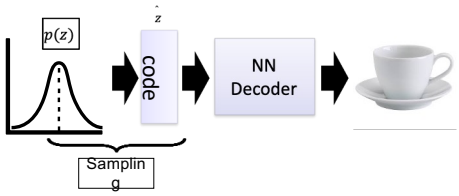
18

Generative Modeling: Variational Autoencoders

Learn a representation by compressing an image as much as possible and then reconstructing it, and push the distribution of the latent representation (i.e z) to a known prior $p(z) \sim N(0, I)$



Images can now be generated from a known prior distribution $p(z)$!

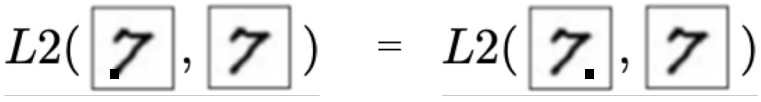


19

19

Generative Modeling: Variational Autoencoders

Unfortunately, the L2 loss does not capture realism



More realistic

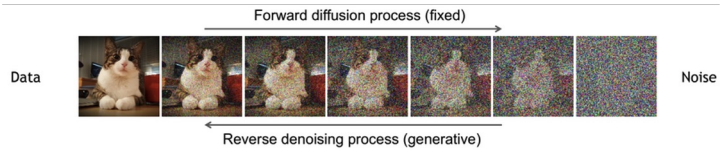
Less realistic

20

20

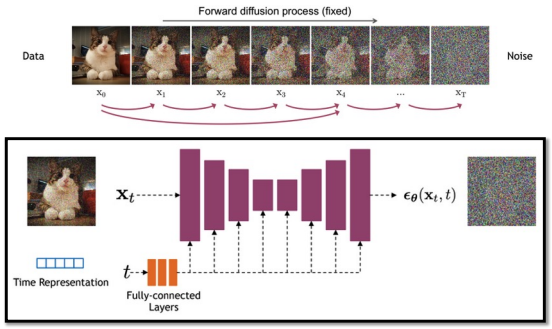
Generative Modeling: Denoising Diffusion models

A Denoising Diffusion model consists of two processes:
- A Forward diffusion process that gradually adds noise to the input
- A Reverse diffusion process that learns to generate data by denoising
- The objective this minimizes is very similar to that of a VAE i.e a variational lower bound



Slides are mostly from the workshop on Denoising Diffusion models, CVPR2022: https://drive.google.com/file/d/1DYHDbt1tSl9oqm3O333biRYzSCOIdtmn/view

Generative Modeling: Denoising Diffusion models, The forward process



Use a U-Net to predict the noise at each step
- Push the noise to be as close to gaussian as possible

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

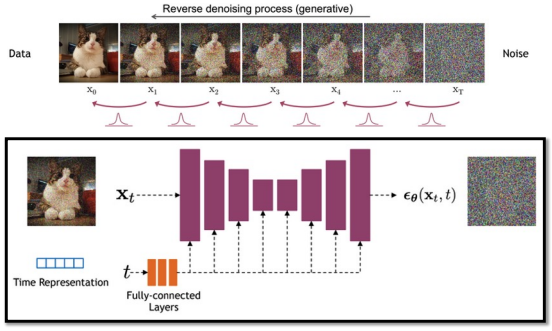
$$\epsilon \sim N(0, \mathbf{I})$$

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

Slides are mostly from the workshop on Denoising Diffusion models, CVPR2022: https://drive.google.com/file/d/1DYHDbt1tSl9oqm3O333biRYzSCOIdtmn/view

Generative Modeling: Denoising Diffusion models, The forward process



Use the predicted noise by U-Net to invert the image

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

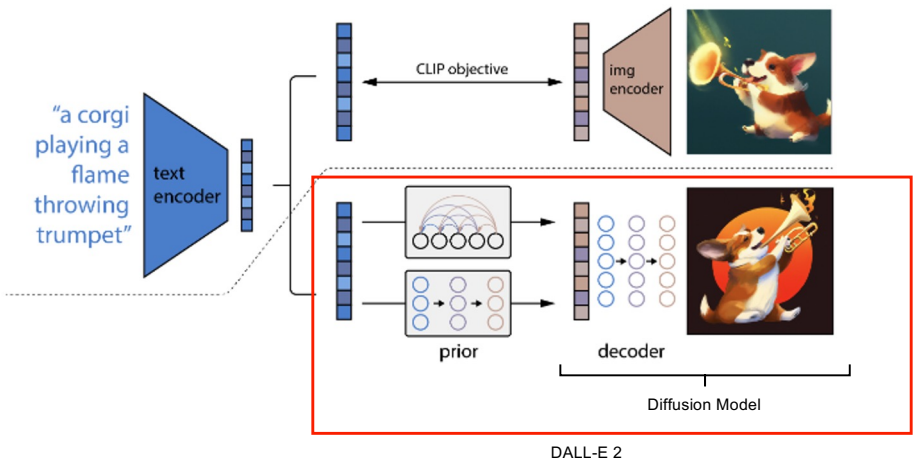
$$\epsilon \sim N(0, \mathbf{I})$$

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Slides are mostly from the workshop on Denoising Diffusion models, CVPR2022: https://drive.google.com/file/d/1DYHDbt1tSl9oqm3O333biRYzSCOIdtmn/view

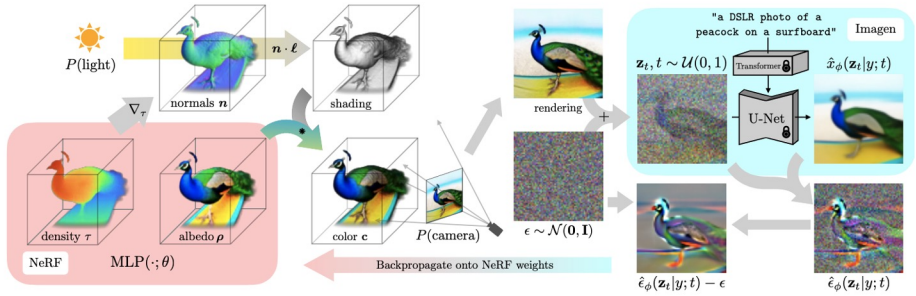
Generative Modeling: DALL-E 2



DALL-E 2

Slides are mostly from the workshop on Denoising Diffusion models, CVPR2022: https://drive.google.com/file/d/1DYHDbt1tSl9oqm3O333biRYzSCOIdtmn/view

Generative Modeling: Text to 3D, DreamFusion

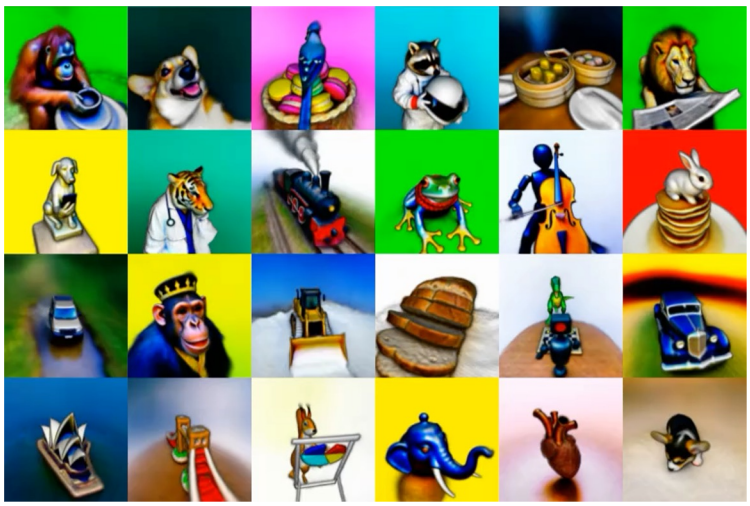


Slides are mostly from the workshop on Denoising Diffusion models, CVPR2022: https://drive.google.com/file/d/1DYHDbt1tSl9oqm3O333biRYzSCOIdtmn/view

25

25

Generative Modeling: Text to 3D, DreamFusion



Slides are mostly from the workshop on Denoising Diffusion models, CVPR2022: https://drive.google.com/file/d/1DYHDbt1tSl9oqm3O333biRYzSCOIdtmn/view

26

26

Generative Modeling: Some Resources

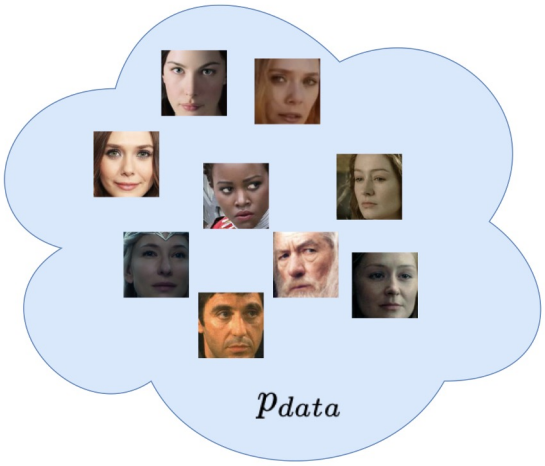
- The Deep Learning Book discusses Autoencoders along with many other generative models: https://www.deeplearningbook.org/
- Durk Kingma's thesis is a great read for probabilistic generative models and VAEs: https://pure.uva.nl/ws/files/17891313/Thesis.pdf
- Lilian Weng's blog post on the VAE is great! https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html
- Diffusion models: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
- Diffusion Models: https://cvpr2022-tutorial-diffusion-models.github.io/

27

27

Generative Adversarial Networks (Goodfellow et al. NIPS2014)

How does one sample from a distribution given only its samples?

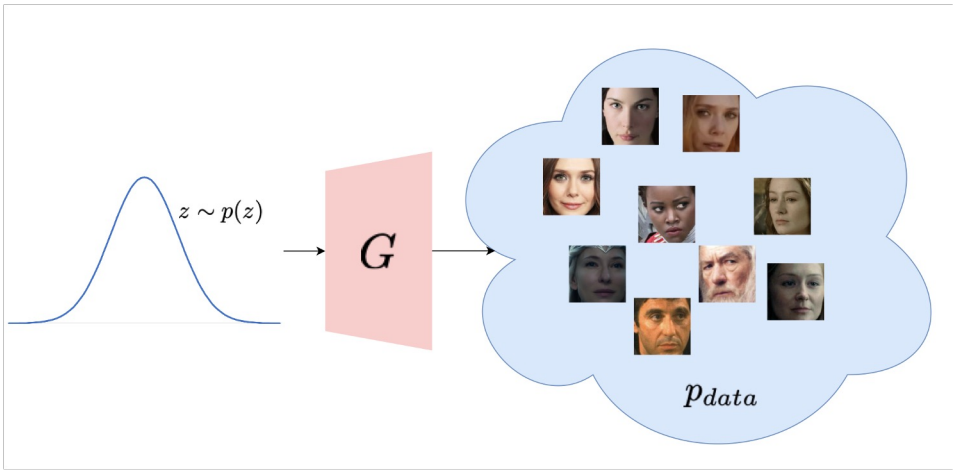


28

28

Generative Adversarial Networks

Train a “Generator” to transform samples from a gaussian to the image space!

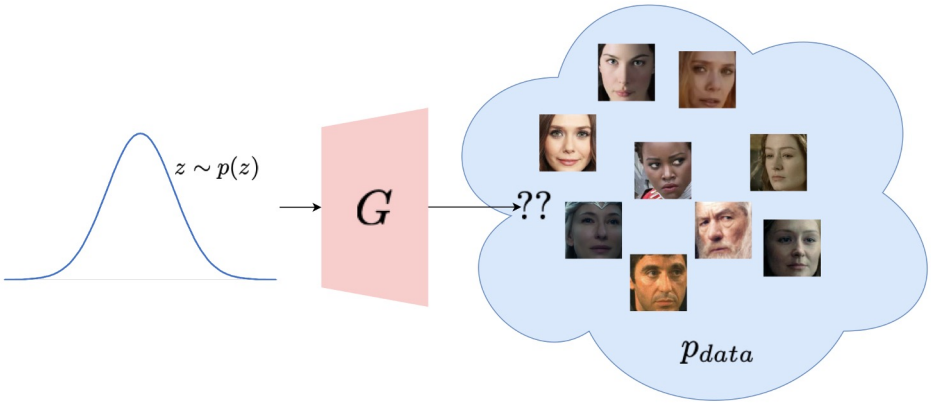


29

29

Generative Adversarial Networks

- But we don’t know what a random z maps to in the Image space, in other words we do not have $(z, Image)$ pairs.
- Supervised losses (L2, L1 etc) cannot be used!

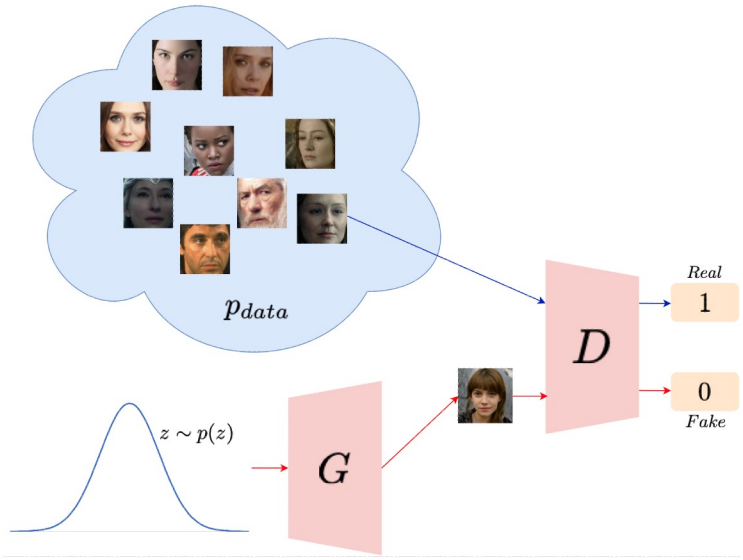


30

30

Generative Adversarial Networks

Use a "Discriminator" to distinguish between samples generated by G and the real samples.

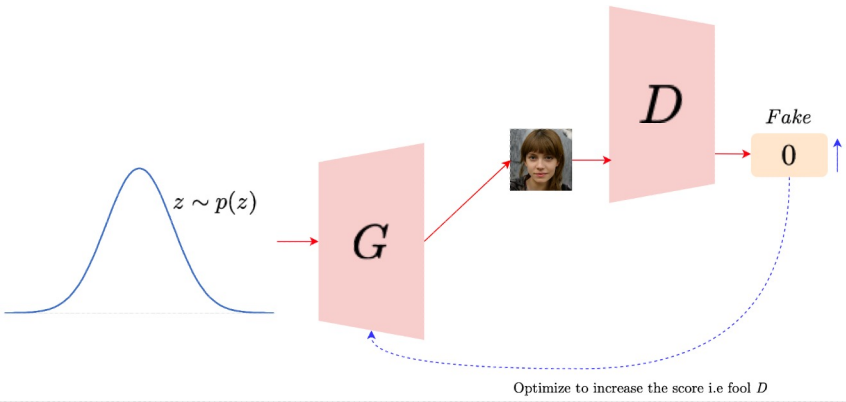


31

31

Generative Adversarial Networks

Train G to maximize D's score. Use the gradient as a "guide" to generate realistic images.

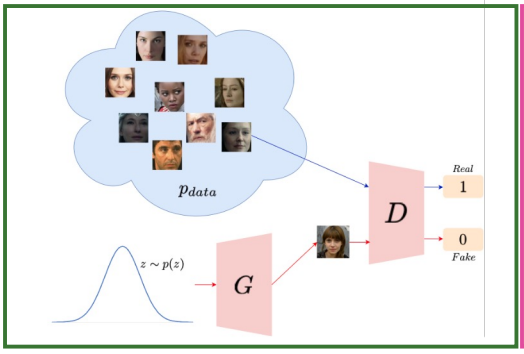


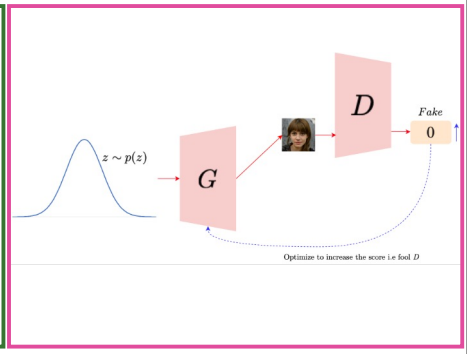
32

32

Generative Adversarial Networks

More formally





$$\max_D \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{z \sim p(z)} [\log (1 - D(G(z)))]$$

Train D to discriminate

$$\min_G \mathbb{E}_{z \sim p(z)} [\log (1 - D(G(z)))]$$

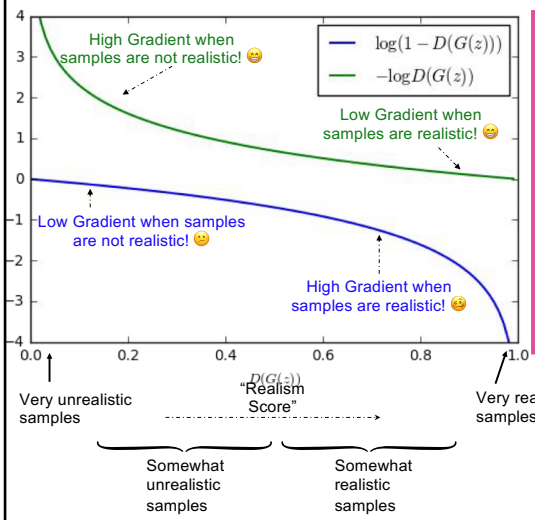
Train G to fool D

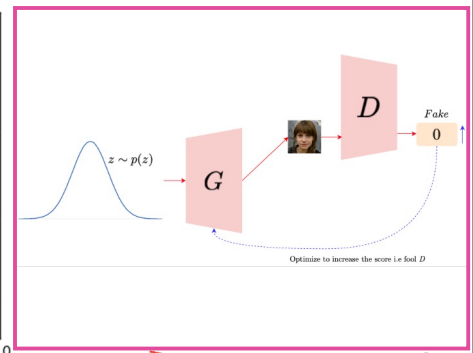
33

33

Generative Adversarial Networks

Fixing the gradient





$$\min_G \mathbb{E}_{z \sim p(z)} [\log (1 - D(G(z)))]$$

Bad Gradients!

$$\max_G \mathbb{E}_{z \sim p(z)} [\log D(G(z))]$$

Better Gradients!

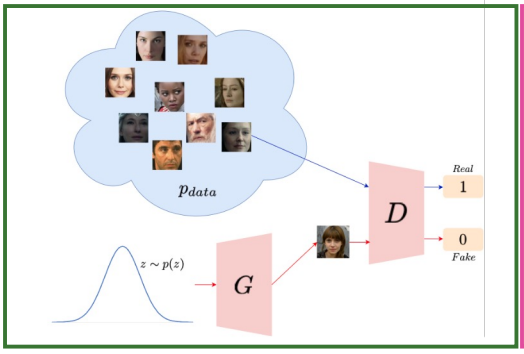
Train G to fool D

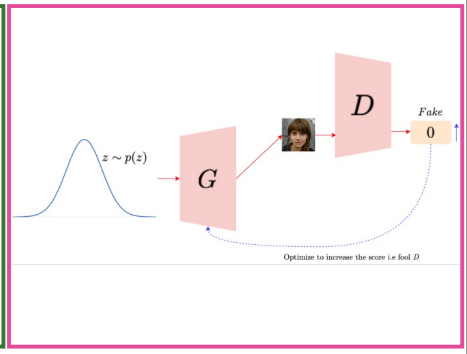
34

34

Generative Adversarial Networks

Finally, the "Non-Saturating" GAN-loss





Train G to fool D

$\max_G \mathbb{E}_{z \sim p(z)} [\log D(G(z))]$

Train D to discriminate

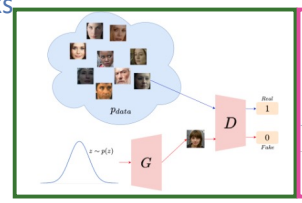
Train G to fool D

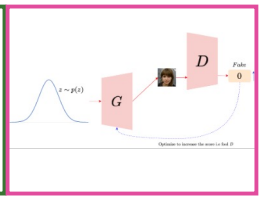
35

35

Generative Adversarial Networks

What's really happening



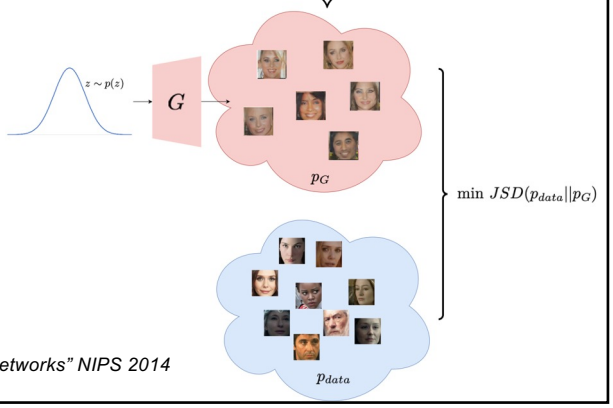


$\max_D \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{z \sim p(z)} [\log (1 - D(G(z)))]$

$\max_G \mathbb{E}_{z \sim p(z)} [\log D(G(z))]$

$JSD(p_{data} || p_G)$ = JSD between p_{data} and p_G.

GAN training minimizes it.



Goodfellow et al. "Generative Adversarial Networks" NIPS 2014

36

Generative Adversarial Networks

GAN samples have become really good. Samples from StyleGAN V2 (Karras et al. CVPR2020)

37

37

Generative Adversarial Networks

GAN samples have become really good. Samples from StyleGAN V2 (Karras et al. CVPR2020)

38

38

Generative Adversarial Networks

GAN samples have become really good. Samples from BigGAN (Brock et al. ICLR2019)

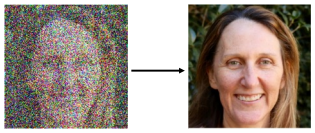
39

39

Generative Adversarial Networks: What can they be used for?

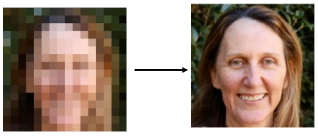
Let's see how GANs can help us for two standard CV problems: *Denoising* and *Super-Resolution*

Denoising



Noisy Image *Real Image*

Super-Resolution : 4×



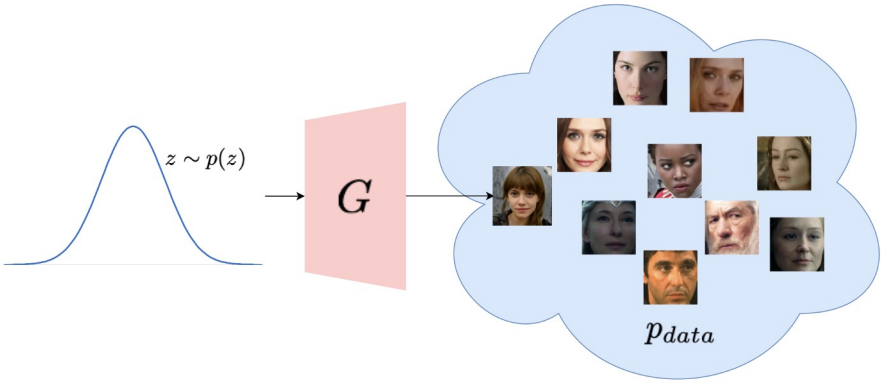
Low-Res Image 32×32 *Real Image* 128×128

40

40

Generative Adversarial Networks

GANs allow us to go from one space to another without using paired data! Here, we're going from vectors to Images!

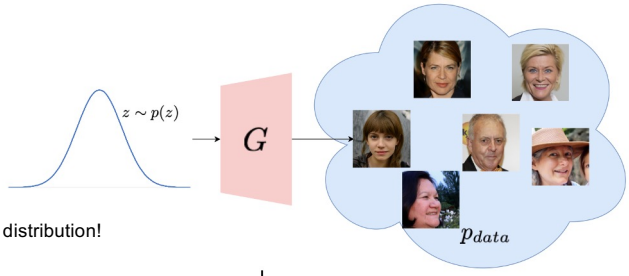


41

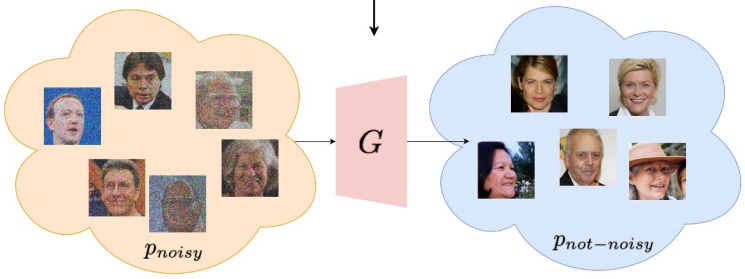
41

GANs for X: X = "Denoising"

Consider the task of going from ***Noisy Images*** → ***Denoised Images***. How can GANs help?



Change the sampling distribution!

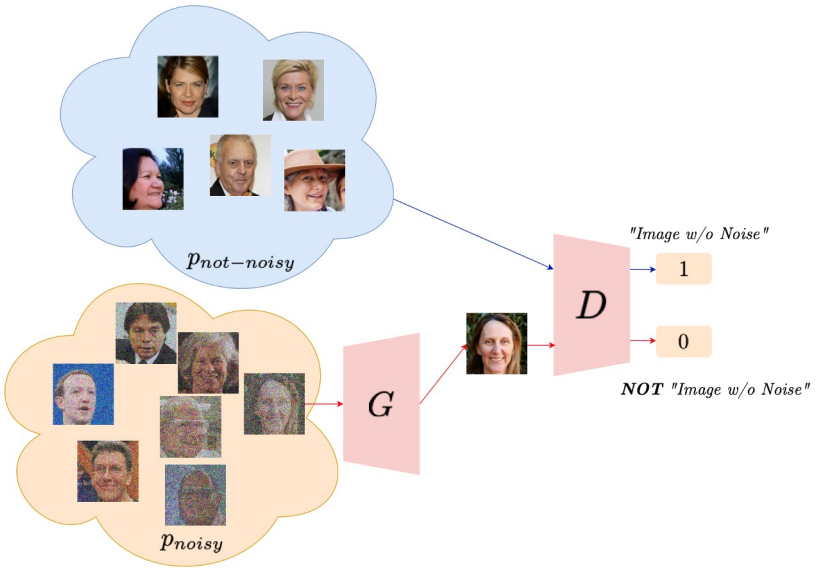


42

42

GANs for X: X = "Denoising"

Train with the adversarial loss! We don't need pairs!

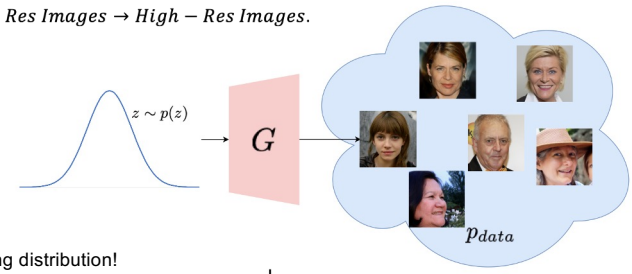


43

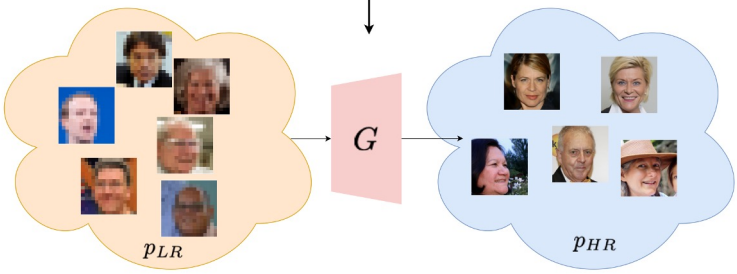
43

GANs for X: X = "Super-Resolution"

Similarly for $Low - Res\ Images \rightarrow High - Res\ Images$.



Change the sampling distribution!

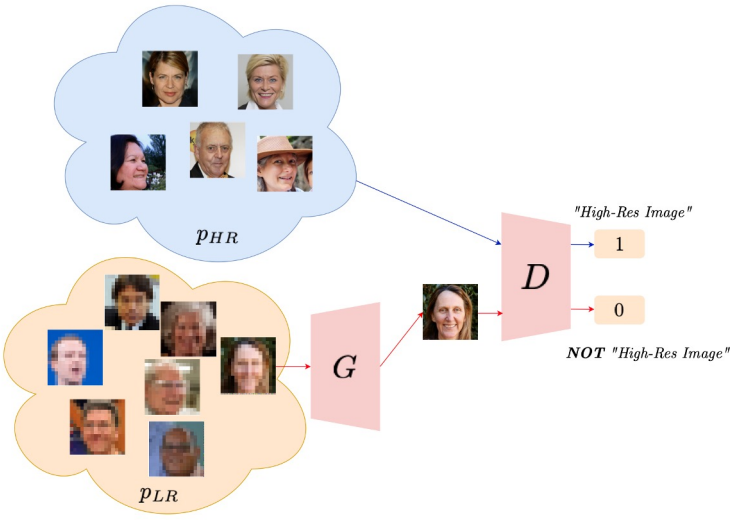


44

44

GANs for X: X = "Super-Resolution"

Train with the adversarial loss! We don't need pairs!

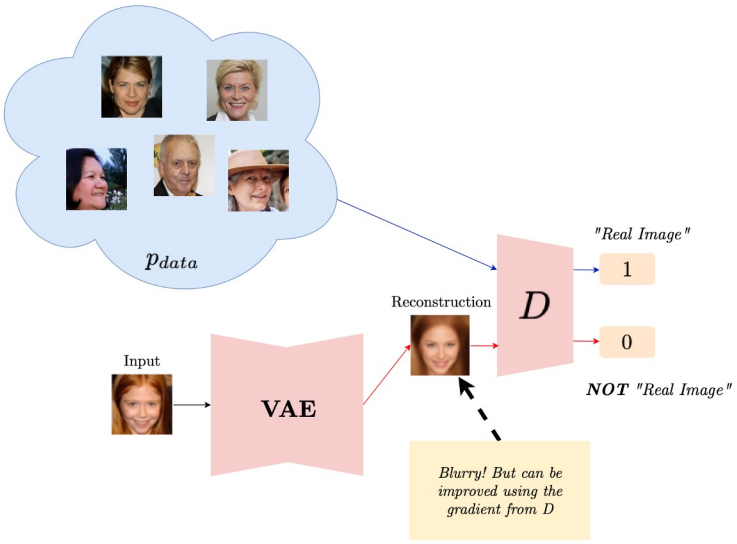


45

45

The Adversarial Loss

Use a discriminator to push VAE reconstructions to be sharper!



Larsen et al. "Autoencoding beyond pixels using a learned similarity metric" PMLR 2016

46

46

The Adversarial Loss

Use a discriminator to push VAE reconstructions to be sharper!

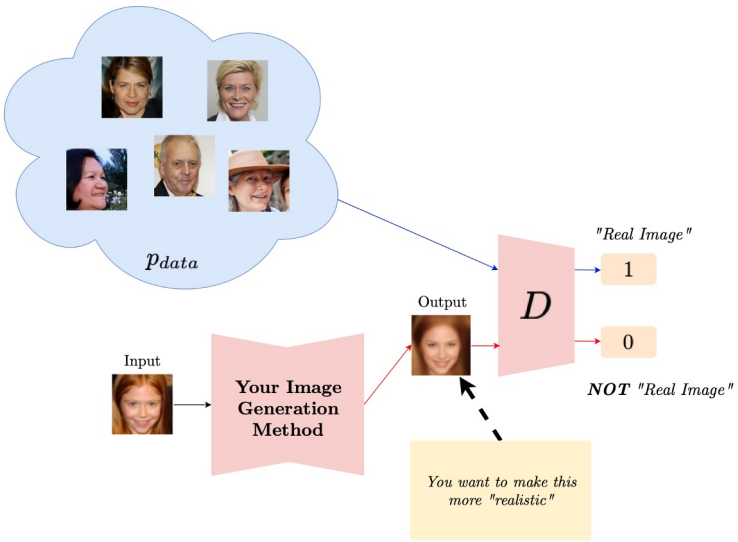
Larsen et al. "Autoencoding beyond pixels using a learned similarity metric" PMLR 2016

47

47

The Adversarial Loss

A discriminator can be used to improve to results of any image generation method!

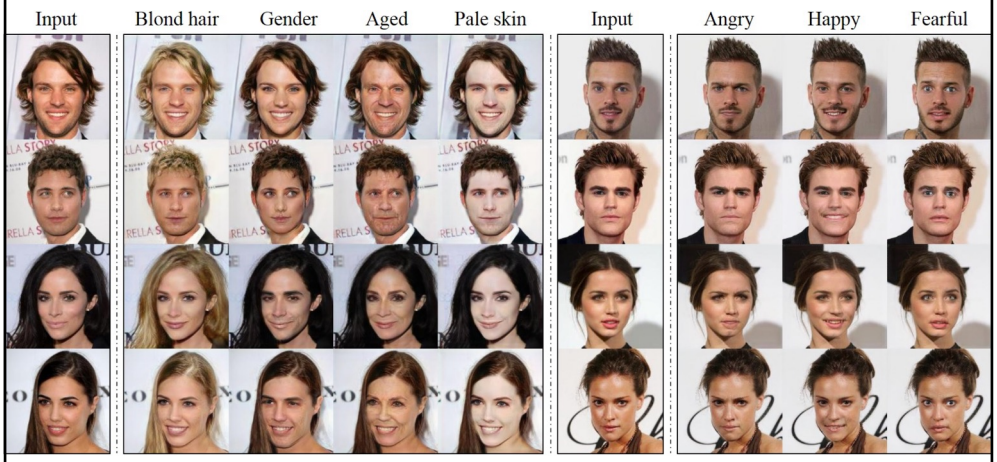


You et al. "Your Paper Tilte" WhereverYouSubmit 2XXX

48

48

StarGAN: GAN for domain translation



49

49

Generative Adversarial Networks: Losses Galore!

The Standard GAN Loss can be pretty unstable to train with. Consequently, multiple losses and regularizers have been proposed

- WGAN-GP (Gulrajani et al. 2017)

$$\underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}.$$

- Least-Squares GAN (Mao et al. 2016)

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[(D(\boldsymbol{x})-1)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z})))^2\right]$$
$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}))-1)^2\right].$$

- Spectral Normalization (Miyato et al. 2018)
- R1 Regularization (Mescheder et al. 2018)

50

50

Generative Adversarial Networks: Architectures

Architectures (including activations, normalizations etc) of both the Generator and the Discriminator are also crucial to generate good samples. Some important work is given below

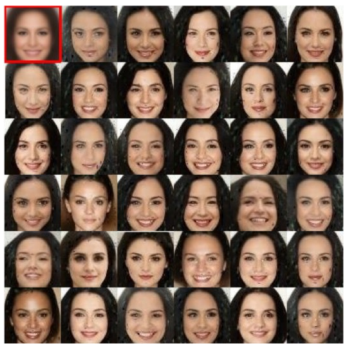
- DCGAN (Radford et al. 2016): First Convolutional Generator
- Pix2pix (Isola et al. 2017): Introduced a patch-based discriminator that discriminated on overlapping patches of the input image.
- Progressive GAN (Karras et al. 2017): First Mega-Pixel level image generation using GANs. Proposed Progressive Training
- CycleGAN (Zhu et al 2017): GANs for image-to-image translation from one domain to another
- StarGAN (Choi et al. 2017): GANs for image-to-image translation between *multiple* domains
- Spectral Normalization (Miyato et al. 2018): First good results on ImageNet classes
- StyleGAN (Karras et al. 2018): High Quality Mega-Pixel level image generation using GANs. Doesn't require progressive training. Introduced a much richer "style" latent space.
- BigGAN (Brock et al. 2019): HQ results in ImageNet Classes

51

51

Generative Adversarial Networks: The problems

- As alluded to earlier, training can be unstable
- Mode Collapse can occur (example from Richardson et al, "On GANs and GMMs", 2018). Faces are very similar!



- Given a sample x, GANs can't tell you how likely (realistic) it is i.e it cannot give you $p_{data}(x)$. The discriminator score cannot be used for this, check out *Goodfellow et al 2014* (the original GAN paper) for why.

52

52

Generative Modeling: Some Resources

- The Deep Learning Book discusses GANs along with many other generative models: Link
- The original GAN paper is a great read!
- Scott Rome's blog digs into the details the the proofs in the GAN paper
- Lilian Weng's blog post on GANs is great too!

53

53

Other Methods are catching up

VQ-VAE-2 (Razavi et al. 2019): Uses a autoregressive prior on the VAE latent space and a much deeper encoder and decoder.

1024×1024

54

54

Other Methods are catching up

GLOW (Kingma et al. 2018): Based on Normalizing Flows (Dinh et al, *NICE: Non-linear Independent Components Estimation*, 2015). Methods based on Normalizing Flows give exact likelihoods $p_{data}(x)$!

256×256

55

55