Introduction to Deep Learning
Chapter 7: Deep Generative Models

Pao-Ann Hsiung

National Chung Cheng University

Contents

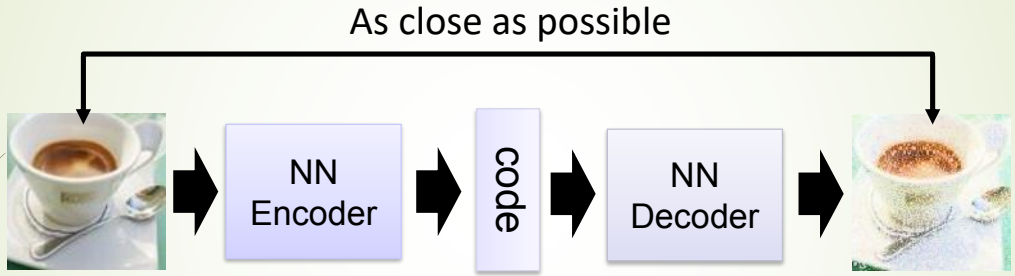
Contents

Introduction

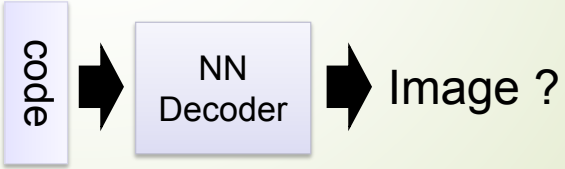
- Why **generation**?
  - Classification (class output) or regression (scalar output) is limited in scope of applications
  - Structured outputs are often required for structured inputs
    - Examples: language translation, text to image, autonomous driving images, etc.

Contents

Autoencoder

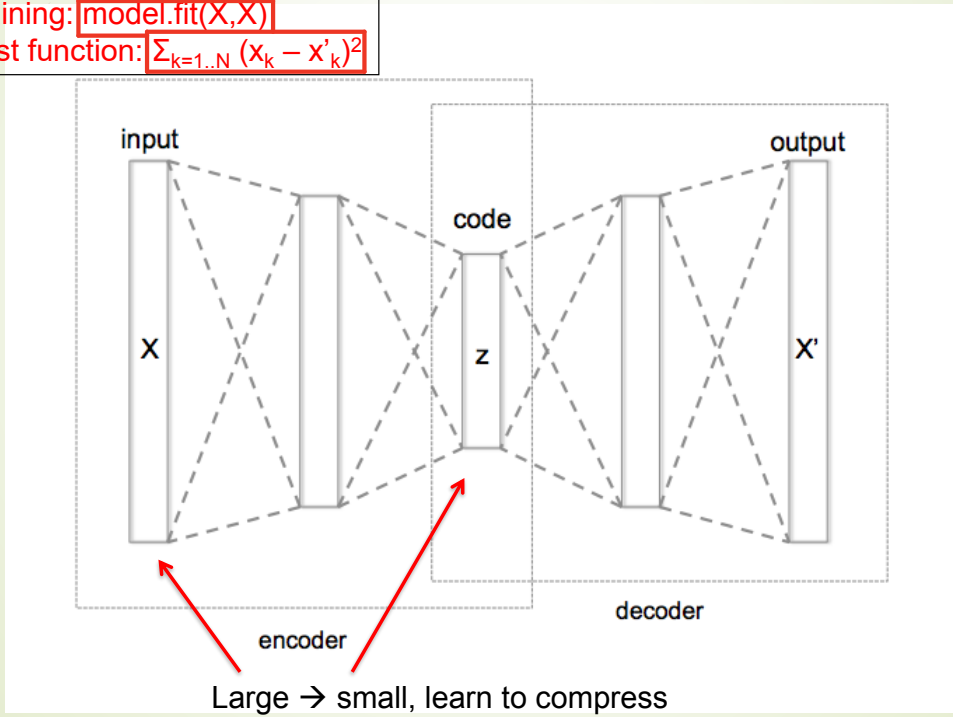


Randomly generate a vector as code

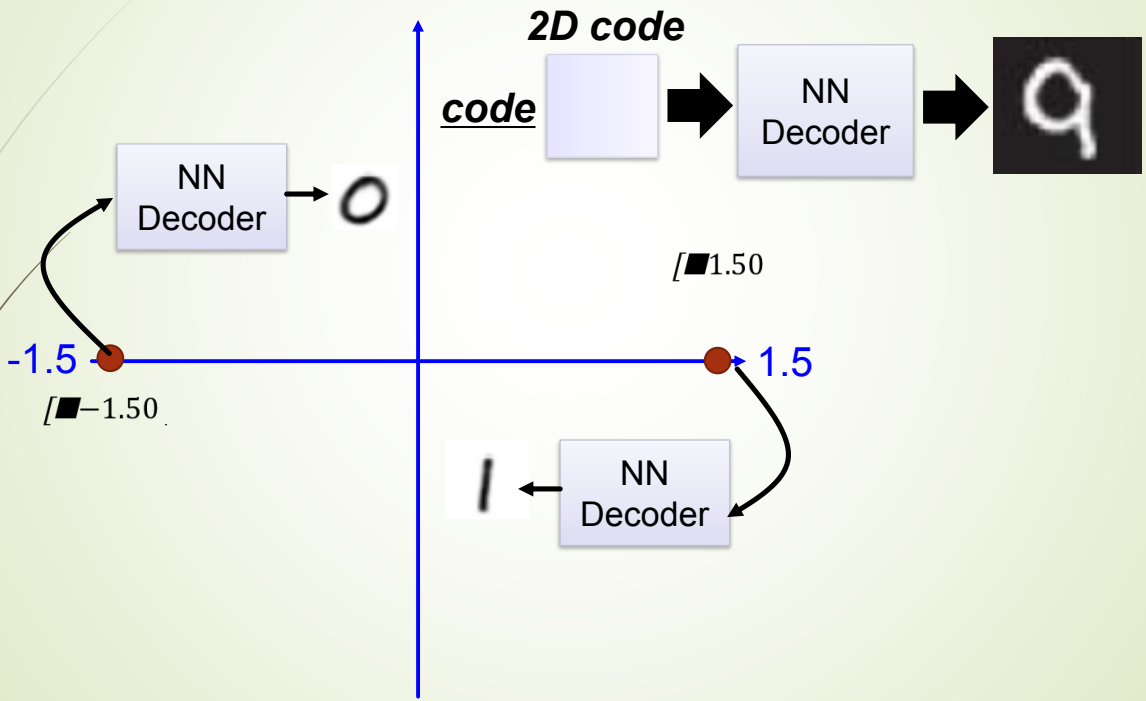


Autoencoder with 3 fully connected layers

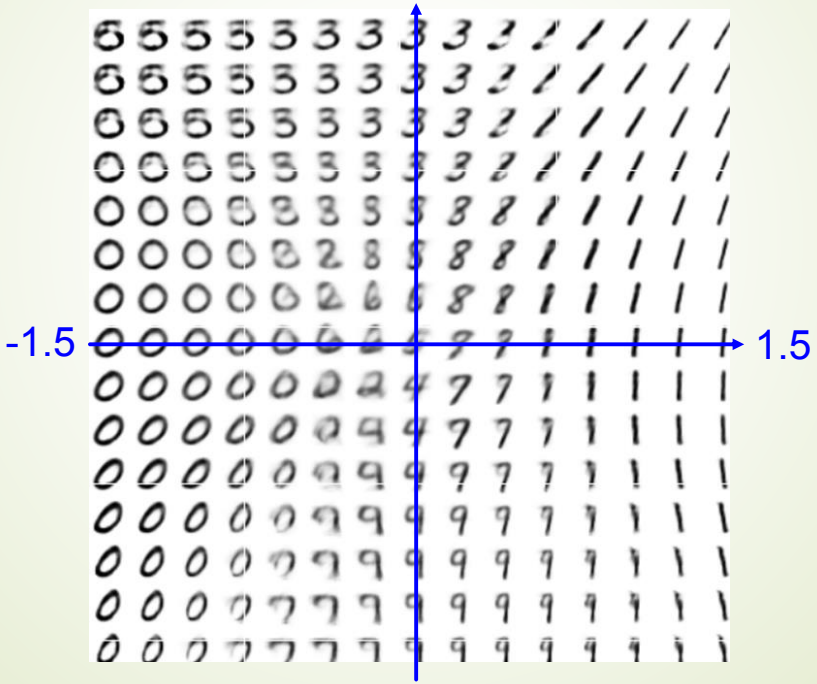
Training: model.fit(X,X)
Cost function: $\Sigma_{k=1..N} (x_k - x'_k)^2$



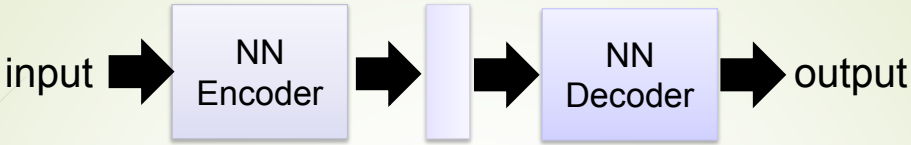
Auto-encoder



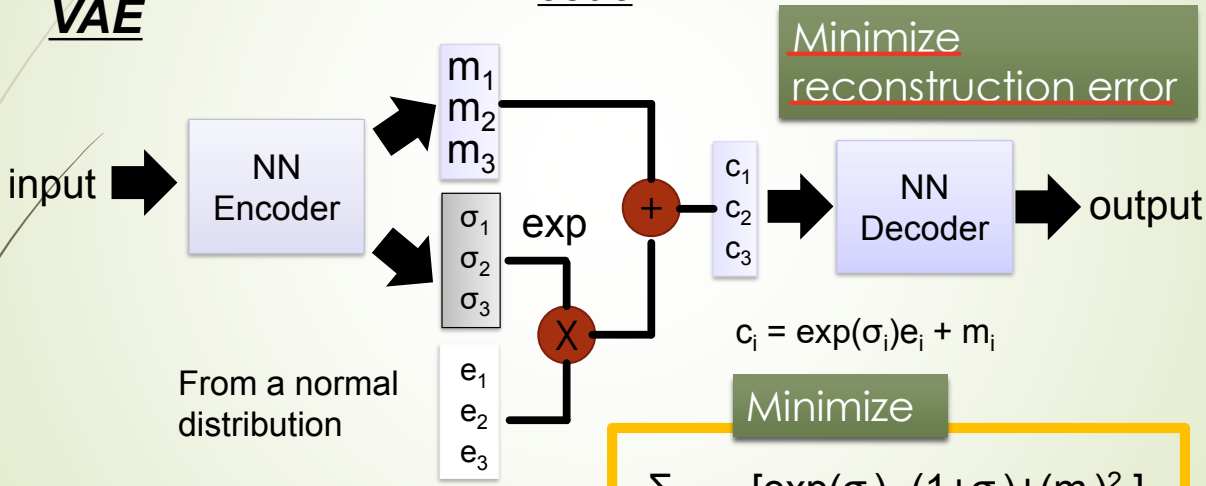
Auto-encoder



Auto-encoder



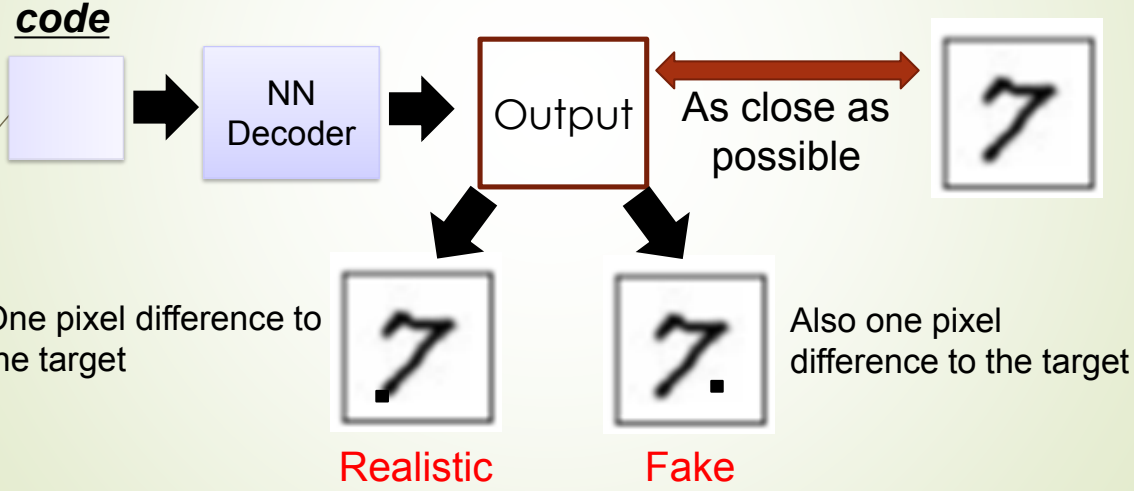
VAE



Auto-Encoding Variational Bayes,
https://arxiv.org/abs/1312.6114

Problems of VAE

- It does not really try to simulate real images

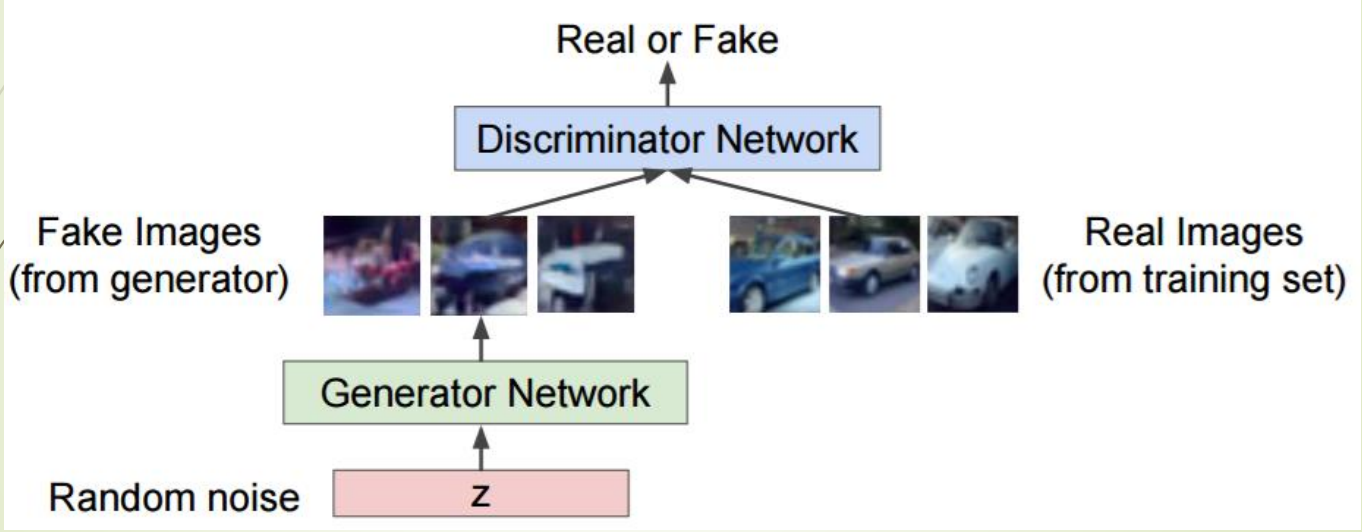


Contents

What are GANs?

- System of two neural networks competing against each other in a zero-sum game framework.
  - Generator and Discriminator
    - dueling each other
- First introduced by Ian Goodfellow *et al.* in 2014.

GAN: Generator & Discriminator

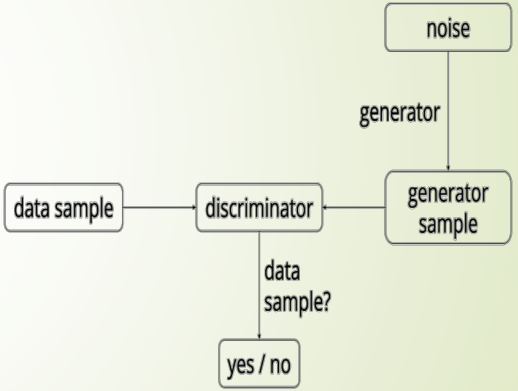


Source: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/generative_models.html

GAN: Generator & Discriminator

Given a set of target samples,
the **Generator** tries to produce samples that can **fool the Discriminator** into believing they are real. The **Discriminator** tries to **distinguish real** (target) samples from **fake** (generated) samples.

After iterative training, we eventually end up with a Generator that is really good at generating samples similar to the target samples.



Overview of GANs
Source: https://ishmaelbelghazi.github.io/ALI

Discriminative Models

- A **discriminative** model learns a function that maps the input data (x) to some desired output class label (y).
- In probabilistic terms, they directly learn the conditional distribution $P(y \mid x)$.

Generative Models

- A **generative** model tries to learn the joint probability of the input data and labels simultaneously i.e. *P(x,y)*.

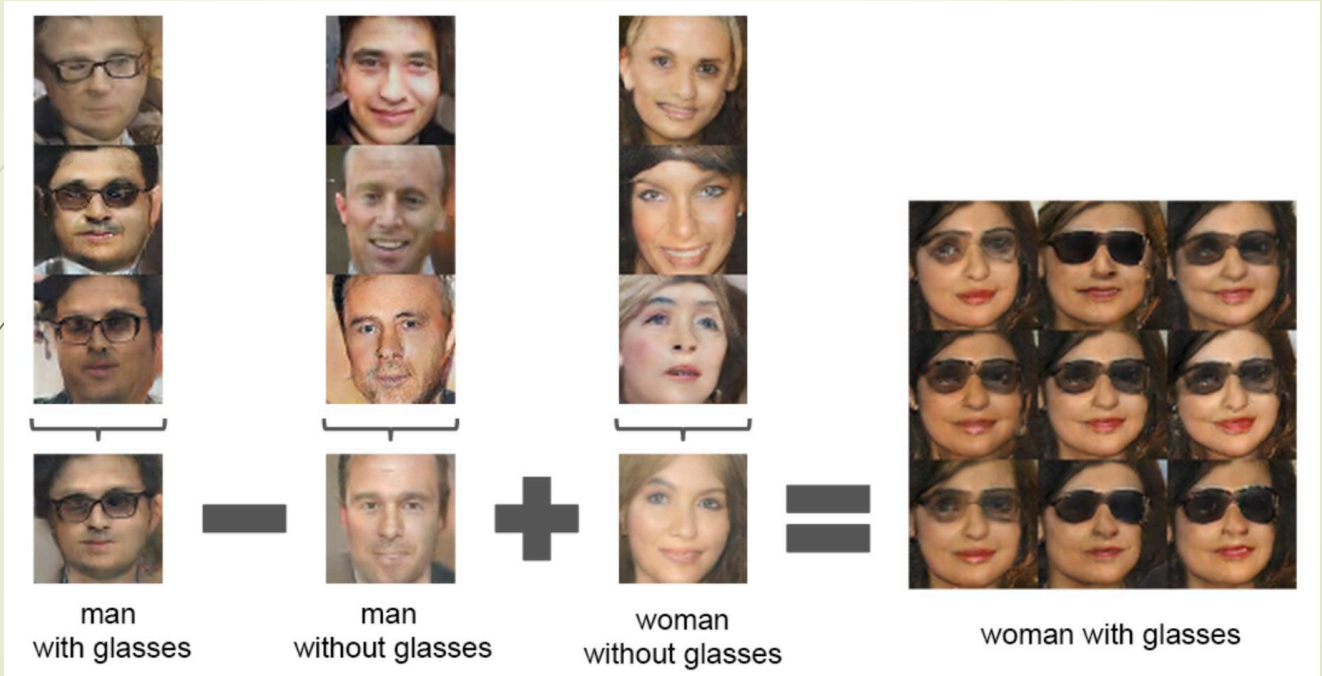
- Potential to understand and explain the underlying structure of the input data even when there are no labels.

How GANs are being used?

- Applied for modelling natural images.
- Performance is fairly good in comparison to other generative models.
- Useful for unsupervised learning tasks.

Why GANs?

- Use a latent code.
- Asymptotically consistent (unlike variational methods) .
- No Markov chains needed.
- Often regarded as producing the best samples.


man
with glasses
man
without glasses
woman
without glasses
woman with glasses

How to train GANs?

Generative與discriminative都需要train

- Objective of generative network - increase the error rate of the discriminative network.
- Objective of discriminative network – decrease binary classification loss.
- Discriminator training - backprop from a binary classification loss.
- Generator training - backprop the negation of the binary classification loss of the discriminator.

否定

Loss Functions

$$\mathcal{L}(\hat{x}) = \min_{x \in data} (x - \hat{x})^2$$

Generator

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

大約1

大約0

Discriminator

https://github.com/hindupuravinash/the-gan-zoo

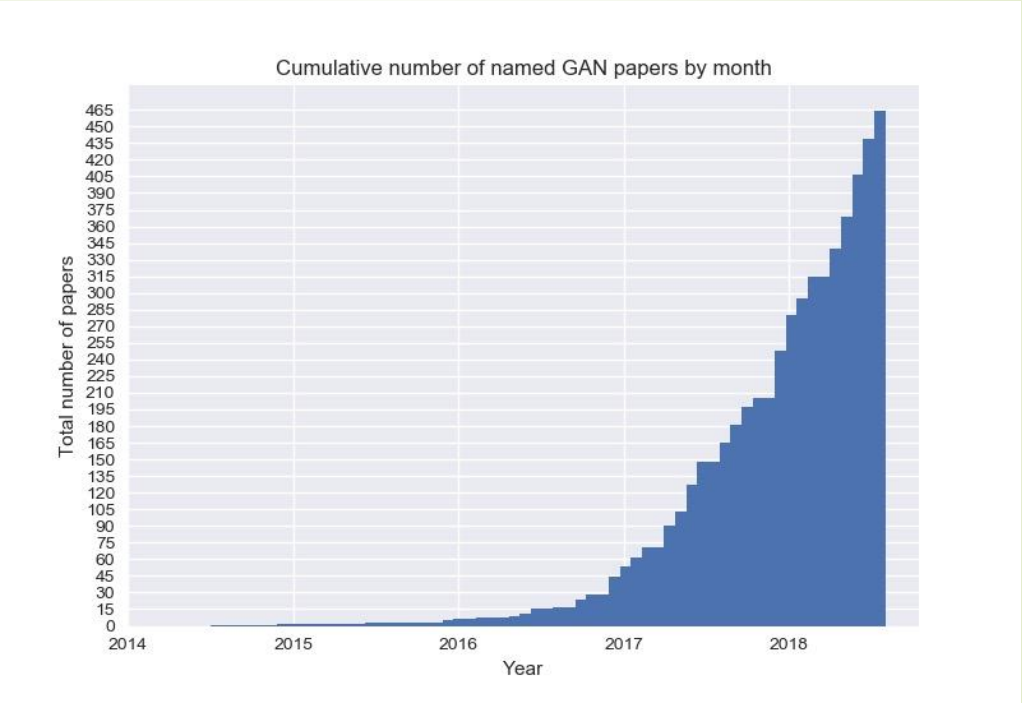
Generated bedrooms. Source: "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks" https://arxiv.org/abs/1511.06434v2

Contents

Variants of GAN

- Look up this website for all current variants of GAN:
- https://github.com/hindupuravinash/the-gan-zoo
- To name a few …
  - 3D-ED-GAN, 3D-GAN, 3d-IWGAN, 3D-PhyNet, 3D-RecGAN, ABC-GAN, AC-GAN, acGAN, ACGAN, AdaGAN, Adaptive GAN, AdvEntuRe, AdvGAN, AE-GAN, AE-OT, AEGAN, AF-DCGAN, AffGAN, AIM, AL-CGAN, ALI, AlignGAN, AlphaGAN, AM-GAN, AmbientGAN, AMC-GAN, AnoGAN, APD, APE-GAN, ARAE, ARIGAN, ArtGAN, ASDL-GAN, ATA-GAN, Attention-GAN, AttGAN, AVID, …

Generative Adversarial Network (GAN)



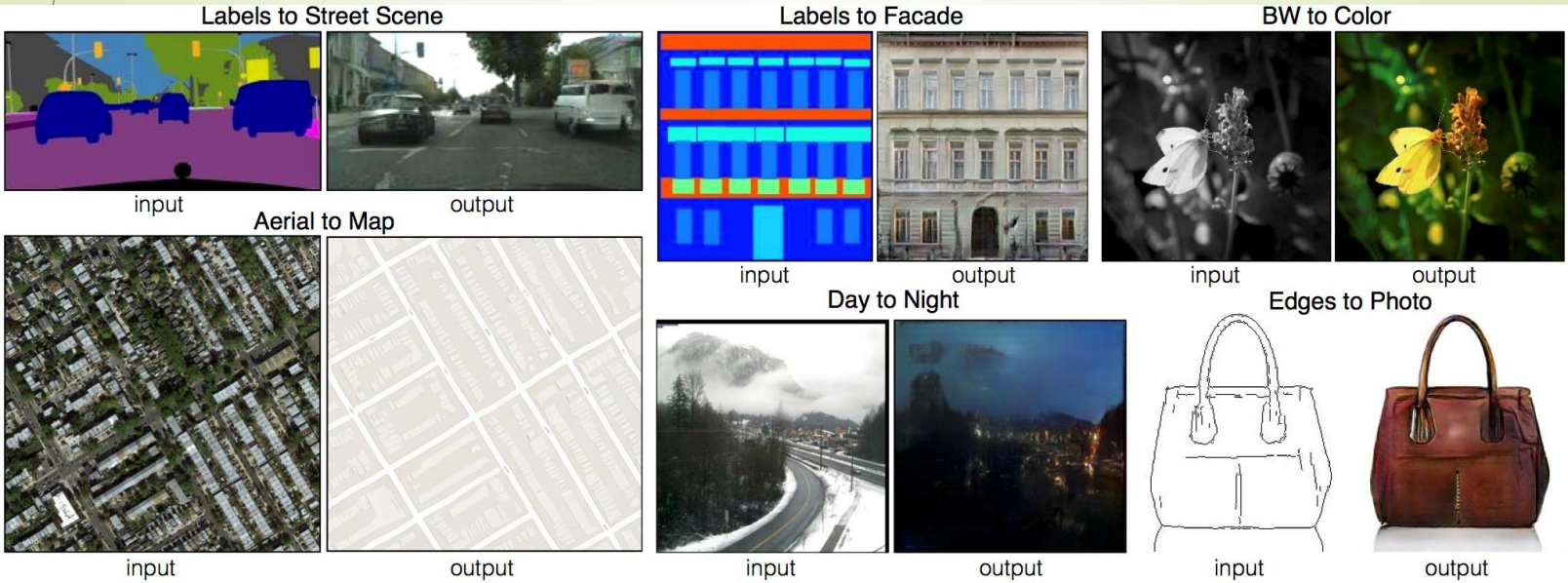
https://github.com/hindupuravinash/the-gan-zoo

Variations to GANs

- Several new concepts built on top of GANs have been introduced –
  - InfoGAN – Approximate the data distribution and learn interpretable, useful vector representations of data.
  - Conditional GANs - Able to generate samples taking into account external information (class label, text, another image). Force G to generate a particular type of output.

Contents

Applications of GAN



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

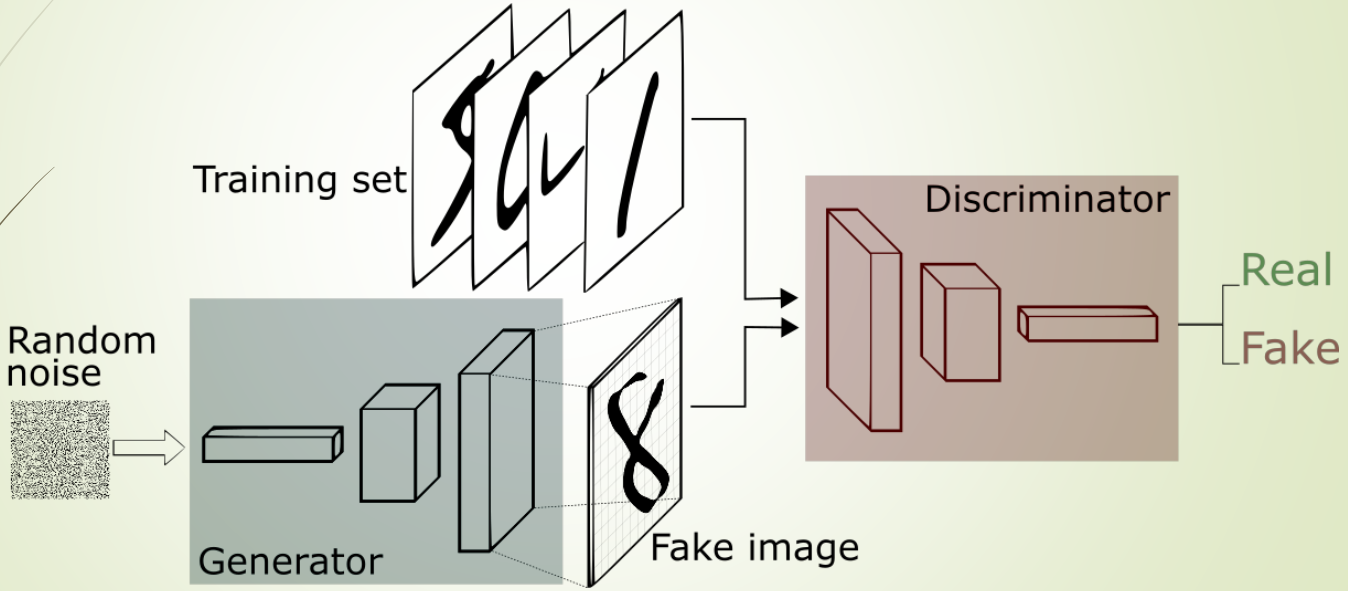
Applications of GAN

- Create characters (Anime)
- Pose Guided Person Image Generation
- Cross-domain transfer
- Clothing creation from images and styles
- Super resolution
- High-resolution image synthesis
- Text to image
- Face synthesis (aging)
- Image inpainting
- Image to image translation (satellite image into map)
- Creating Emoji
- Texture synthesis
- Image (photo) editing
- Video (music) generation

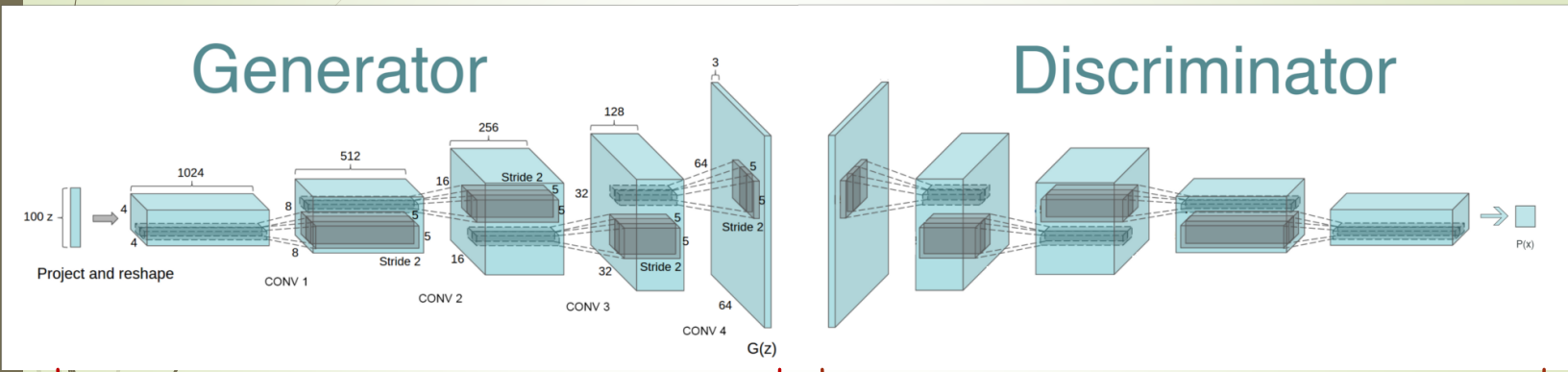
https://medium.com/@jonathan_hui/gan-some-cool-applications-of-gans-4c9ecca35900

Contents

Generative Adversarial Network (GAN)



Generative Adversarial Network (GAN)

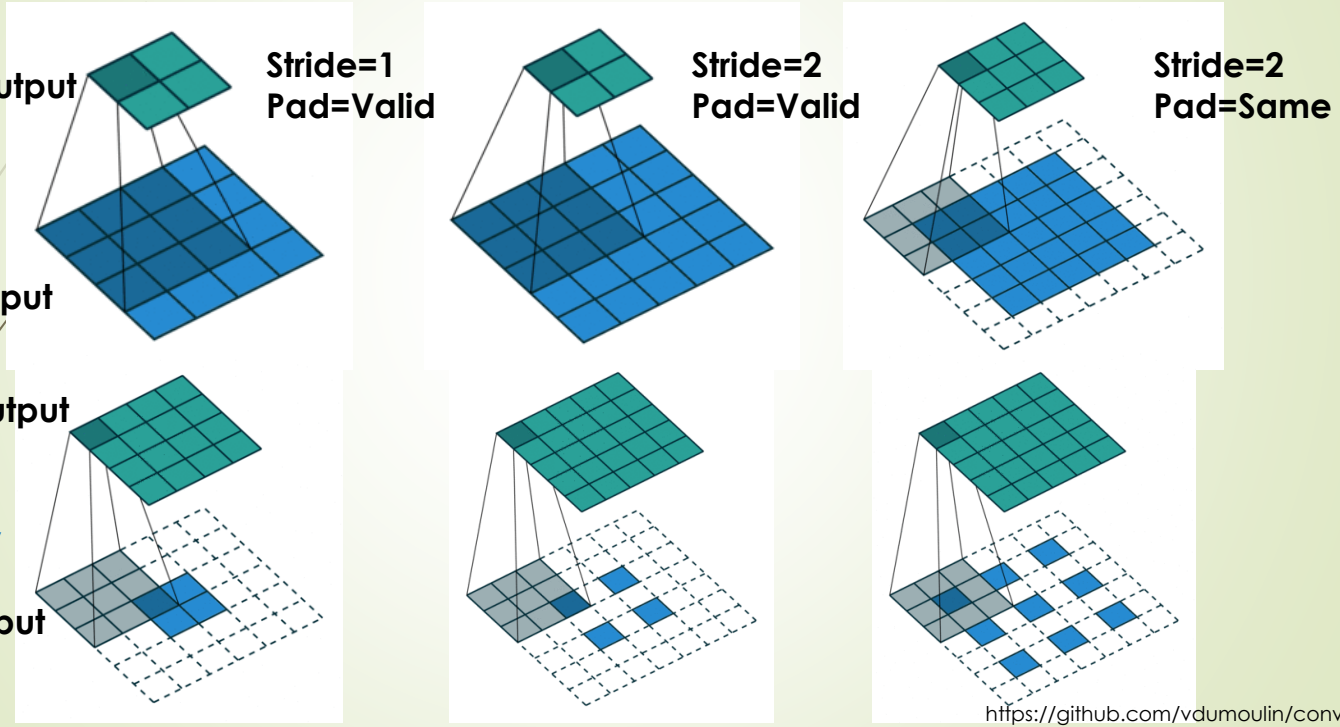


New components:
Transposed convolution,
Batch Normalization

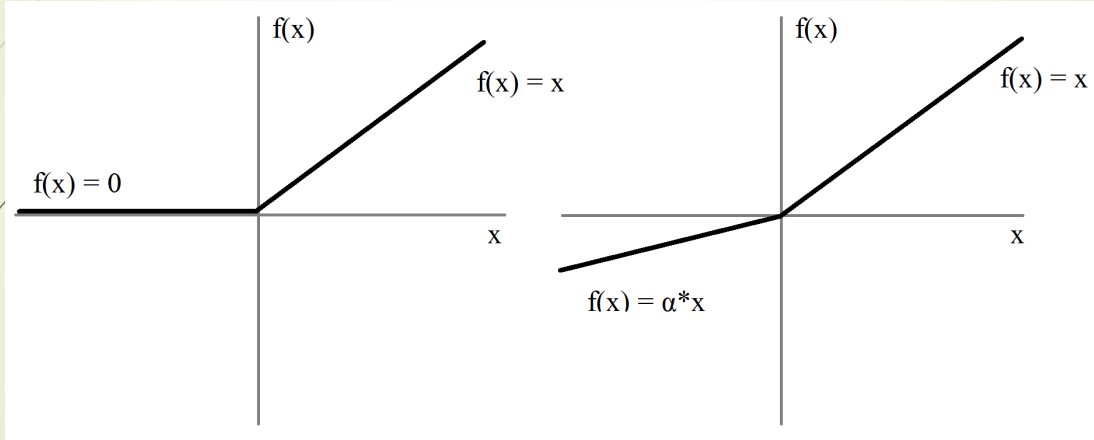
Binary Classifier:
Conv, Leaky ReLU,
FC, Sigmoid

https://github.com/PramodShenoy/GANer

Transposed Convolution (Deconvolution)



ReLU vs. Leaky ReLU (LReLU)



Used in Generator **Used in Discriminator**

Batch Normalization (BN)

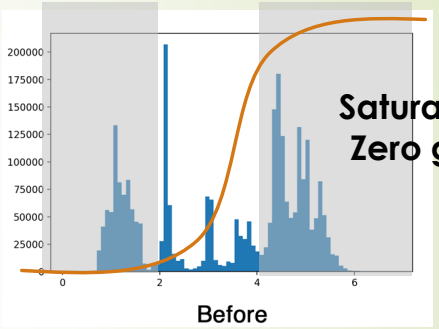
Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

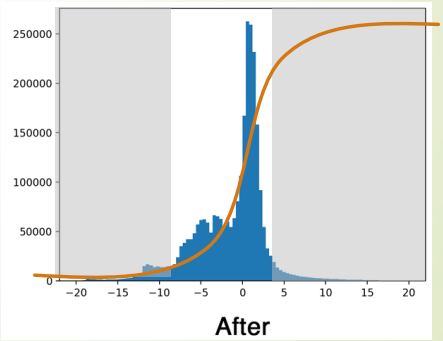
$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

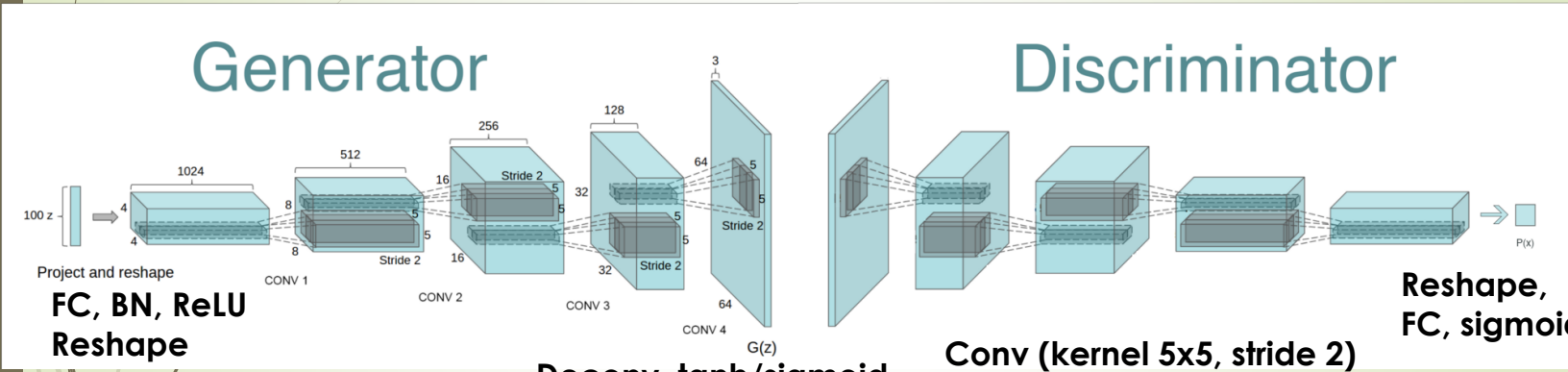


Saturation area
Zero gradient



https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c

Generative Adversarial Network (GAN)



FC, BN, ReLU
Reshape

Deconv (kernel 5x5, stride 2)
BN, ReLU

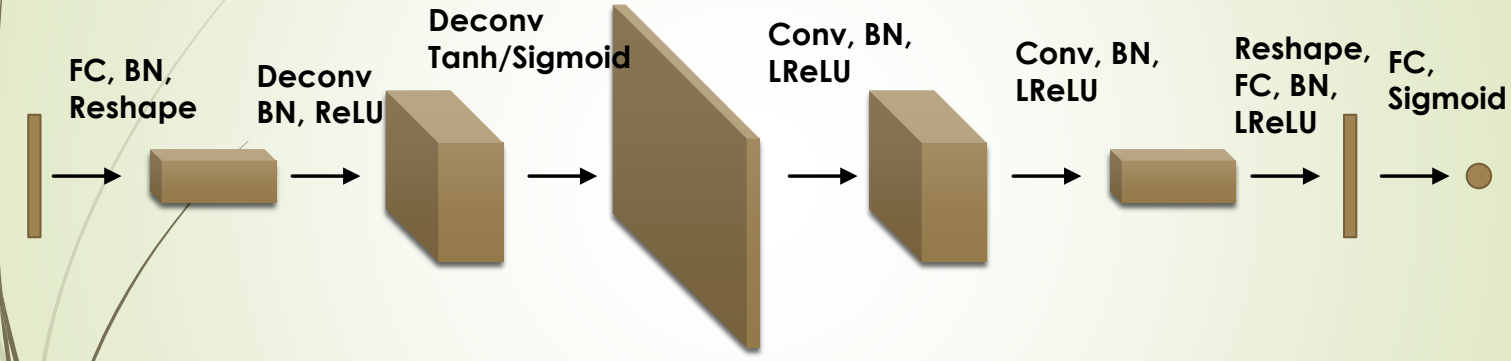
Deconv, tanh/sigmoid

Conv (kernel 5x5, stride 2)
BN, LReLU (slope 0.2)

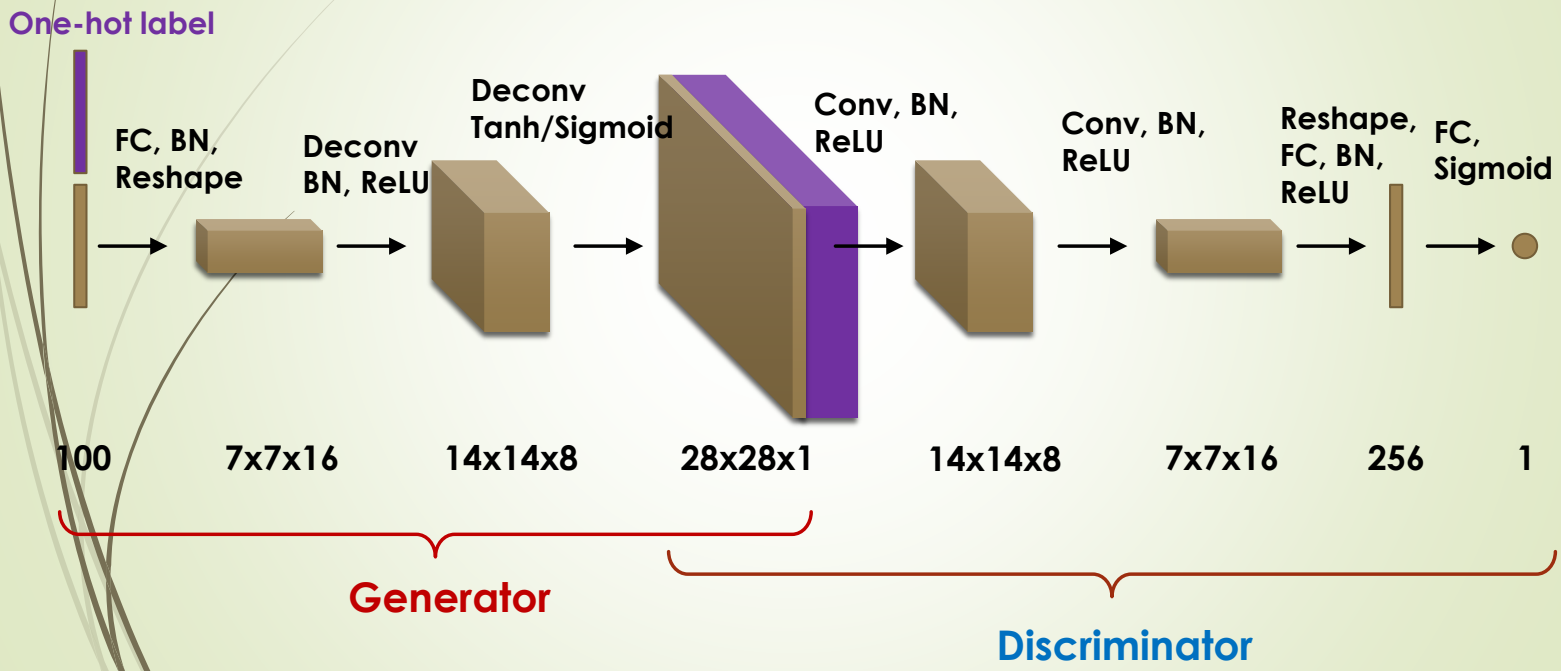
Reshape,
FC, sigmoi

https://github.com/PramodShenoy/GANerations

GAN on MNIST


FC, BN, Reshape
Deconv BN, ReLU
Deconv Tanh/Sigmoid
Conv, BN, LReLU
Conv, BN, LReLU
Reshape, FC, BN, LReLU
FC, Sigmoid
100
7x7x16
14x14x8
28x28x1
14x14x8
7x7x16
256
1
Generator
Discriminator

Conditional GAN on MNIST



TensorFlow Implementation of GAN

Run the demo code: (Linux or MacOS)

$ git clone https://github.com/carpedm20/DCGAN-tensorflow

$ cd DCGAN-tensorflow

$ pip install tqdm (if you do not have this package)

$ python download.py mnist

$ python main.py --dataset mnist --input_height=28 --output_height=28 --train

TensorFlow Implementation of GAN

- Input: z, image, (label)
- Network: D, G
- Loss: D, G
- Optimizer: D, G

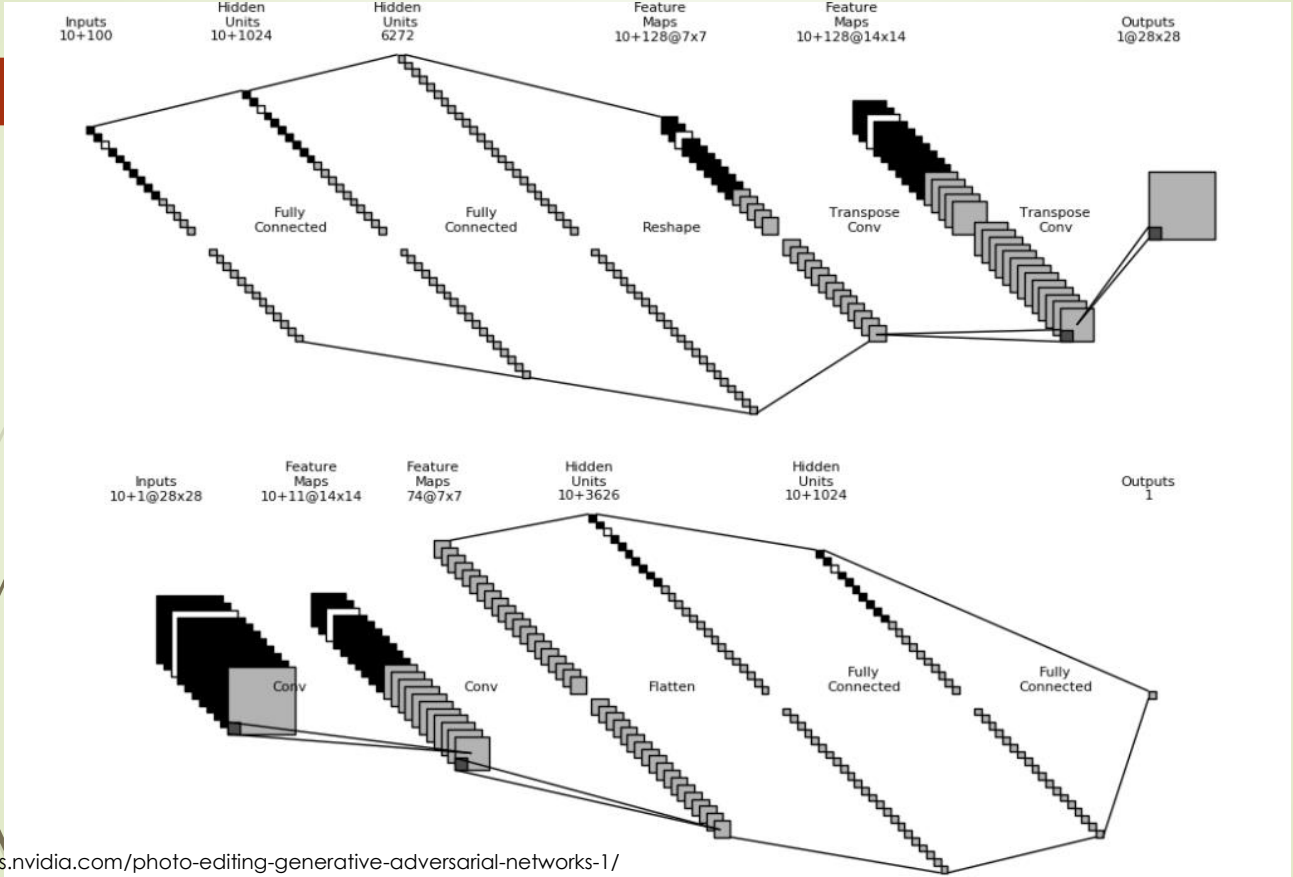
Training:

for epoch

    for batch

        Update D

        Update G

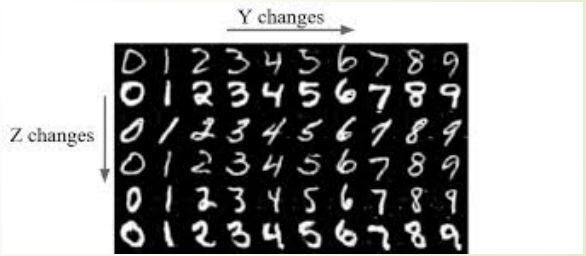


https://devblogs.nvidia.com/photo-editing-generative-adversarial-networks-1/

TensorFlow Implementation of GAN

Testing:

- **Random Generation**
- **Conditional Generation**
- 



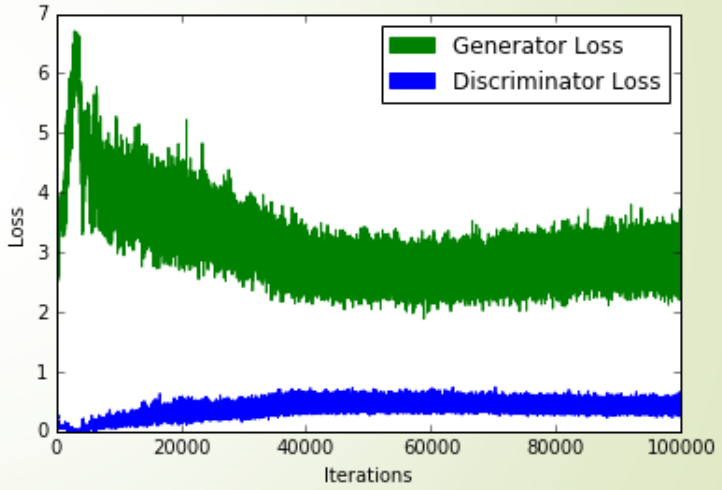
Contents

Major Difficulties

- Networks are difficult to converge.
- Ideal goal – Generator and discriminator to reach some desired equilibrium but this is rare.
- GANs are yet to converge on large problems (E.g. Imagenet).

When to stop training GAN?

- Generator loss improves when Discriminator loss degrades
- Discriminator loss improves when Generator loss degrades

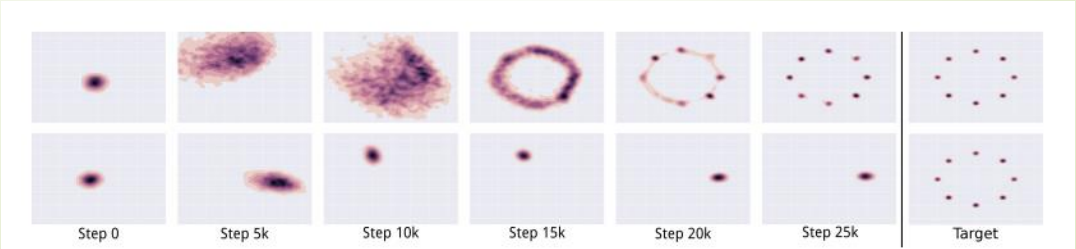


Common Failure Cases

- The discriminator becomes too strong too quickly and the generator ends up not learning anything.
- The generator only learns very specific weaknesses of the discriminator.
- Mode collapse in generator
  - The generator learns only a very small subset of the true data distribution.

Mode Collapse in Generator

- Natural data distributions: multimodal (lots of peaks or modes)
- Generator learns only a limited set of modes to fool the discriminator
  - Generator locks on to a few modes and if the discriminator is not fooled, it switches to some other modes, the cycle repeats!



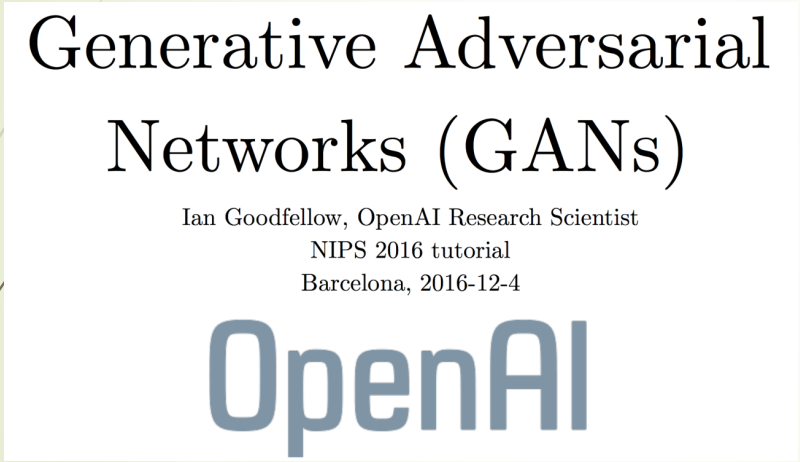
So what can we do?

- Normalize the inputs
- A modified loss function
- Use a spherical Z
- BatchNorm
- Avoid Sparse Gradients: ReLU, MaxPool
- Use Soft and Noisy Labels
- DCGAN / Hybrid Models
- Track failures early (D loss goes to 0: failure mode)
- If you have labels, use them
- Add noise to inputs, decay over time

Conclusions

- Train GAN – Use discriminator as base model for transfer learning and the fine-tuning of a production model.
- A well-trained generator has learned the true data distribution well - Use generator as a source of data that is used to train a production model.

Dive Deeper?



Ian Goodfellow's NIPS 2016 Tutorial

Available online.

References

- https://tryolabs.com/blog/2016/12/06/major-advancements-deep-learning-2016/
- https://blog.waya.ai/introduction-to-gans-a-boxing-match-b-w-neural-nets-b4e5319cc935#.6l7zh8u50
- https://en.wikipedia.org/wiki/Generative_adversarial_networks
- http://blog.aylien.com/introduction-generative-adversarial-networks-code-tensorflow/
- https://github.com/soumith/ganhacks
- Some cool applications of GAN: https://medium.com/@jonathan_hui/gan-some-cool-applications-of-gans-4c9ecca35900
- Implementation of GAN: https://docs.google.com/viewer?url=http%3A%2F%2Fweb.eecs.utk.edu%2F~qi%2Fdeeplearning%2Flecture13-gan-implementation.pptx
- Advances in GAN (including issues and possible solutions) https://medium.com/beyondminds/advances-in-generative-adversarial-networks-7bad57028032