Tutorial on Generative Adversarial Networks - From basics to current state-of-the-art, and towards key applications in medicine

Răzvan V. Marinescu

Medical Vision Group, Massachusetts Institute of Technology

Slides available online: https://people.csail.mit.edu/razvan
Generative adversarial networks

- GAN basics
- State-of-the-art
- Key applications in computer vision and medicine
- Preliminary results on three different medical datasets
Introduction to Generative Models

- Generative: can generate new data instances

  \[ p(X, Y) \]

- Discriminative: discriminates between different kinds of data instances

  \[ p(Y|X) \]

\[ X = \text{image} \]
\[ Y = \text{label/score} \]
Introduction to Generative Adversarial Networks

- Introduced by Goodfellow et al, 2014

- Two adversaries (generator + discriminator) compete with each other

- Over time, the generator gets better at generating images
GAN architecture

- Generator attempts to generate good images to fool the discriminator
- Discriminator attempts to tell apart the fake images from the real ones
- Loss function:

\[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z)))]
\]
Towards the state-of-the-art in Generative Adversarial Networks
Initial GAN results (DCGAN, Radford et al, 2015)

- Initial results looked promising, but still a long way from photorealism

- Other problems persisted:
  - Training collapse
  - Mode collapse
  - Low coverage

- Dynamics between G & D not well understood

Standard samples

Training collapse

Mode collapse
Wasserstein GANs (Arjovski et al, 2017)

- Original GANs were very hard to train

- Problem: They optimise the Jensen-Shannon divergence, a “vertical” distance → no good gradients when distributions far away

- A “horizontal” distance (e.g. Wasserstein) ensures gradients are non-zero when distributions don't have support (i.e. far away)

- GAN trained with the new Wasserstein metric collapses less often, and generates better images
Progressive Growing of GANs (Karras et al, 2018)

- GAN training unstable if one starts directly in high-resolution

- Key idea: start from low-resolution (4x4) and build up to highest-resolution (1024x1024)

- Each new layer is faded-in slowly
StyleGAN1 (Karras et al, 2019)

- Borrows ideas from style-transfer literature
- Uses a mapping network to generate “style vectors“ at every level in the generator
- Each style vector is intensity-normalized (AdaIN operation)
- Generated images have unprecedented realism and diversity
The style-based architecture allows both style-transfer and fine-level stochastic variation.
StyleGAN2 improvements (Karras et al, 2019b)

Blob-artefacts caused by AdaIN normalisation

Solution: bake normalisation straight into convolution weights:

\[ w''_{ijk} = \frac{w'_{ijk}}{\sqrt{\sum_{i,k} w'_{ijk}^2 + \epsilon}} \]

“phase” artefacts due to progressive growing

Solution: communicate across resolution levels through skip connections

Razvan V. Marinescu
razvan@csail.mit.edu
https://people.csail.mit.edu/razvan/
Previous StyleGAN2 model needed large number of images for training ($\approx 70,000$)

Aim: Enable training on limited datasets (1000 images) through data-augmentation

Problem: augmentations leak into the generated images

This is mitigated by ensuring augmentation probability $p$ is lower than a threshold ($\approx 0.9$).
Applications of GANs and other generative models
Application 1: Image super-resolution

- \( G \) generates high-res image from low-res input
- \( D \) discriminates whether high-res image is fake or real.

(Ledig et. al., 2017)
Application 2: In-painting

(a) Input context

(b) Human artist

(c) Context Encoder
   \( (L2 \text{ loss}) \)

(d) Context Encoder
   \( (L2 + \text{Adversarial loss}) \)

(Pathak et al, 2016)
Application 3: Image-to-image translation

Zebras ↔ Horses

summer ↔ winter

horse → zebra

winter → summer

CycleGAN (Jun-Yan Zhu, CVPR, 2016)
Applications in medical imaging

MRI super-resolution (Sanchez et al., 2018)

Modality translation (Zhang et al, 2019)

MR Reconstruction from undersampled K-space (Quan et al, 2017, Yang et al, 2017)

MRI motion correction (Usman et al, 2020)

Any image reconstruction task!
Applications of generative models to medicine: prediction of disease progression

- Prediction and visualisation of future of disease progression
- Can assist doctors in assigning treatments

Alzheimer’s disease prediction (Ravi et. al., 2019)

Fig. 4. Neurodegeneration simulation of a 69-year old ADNI participant.
Preliminary results of StyleGAN2 on three medical datasets
Preliminary results of StyleGAN2 on Chest X-rays

- StyleGAN2 out-of-the-box
- Trained on MIMIC III, 360k images of 1024x1024 resolution
- Still some problems to fix:
  - some ribs look "broken"
  - bone contours are not always smooth/straight
Preliminary results of StyleGAN2 on brain MRI

- Still out-of-the-box model (StyleGAN2)
- Trained on 8,000 brain scans (ADNI/OASIS/AIBL/PPMI)
- Next:
  - check if neuroanatomical properties are preserved (e.g. brain/ventricle vols are same)
  - extend StyleGAN2 to 3D
Preliminary results of StyleGAN2 on microscopy images

- Out-of-the-box
- Trained on 11,000 microscopy slices with pancreatic cancer (MICCAI PANDAS 2020 challenge dataset)
- 512x512 image crops
Conclusion

- GANs have obtained state-of-the-art results on image generation
- Can generate sharp, realistic images
- Training now stable compared to 2-3 years ago, but can take up to 6-7 days (StyleGAN2) on 8 GPUs.
- Can help solve image reconstruction tasks

- Many potential applications in medical imaging

- Recommendations:
  - Don’t build your own, start with a state-of-the-art model (StyleGAN2, BigGAN or Karras, 2020)
  - Download models already pre-trained to explore their capabilities
  - When training, initialise weights from another pre-trained model instead of random
  - PIs: Include costs for buying GPUs/AWS-credits in your grant applications

- Keep an eye on other types of generative models (VAEs, auto-regressive, flow) that have other interesting properties (e.g. density estimation), which enable other tasks e.g. anomaly detection