ChatGPT prompt “minimalist landscape painting of a deep underwater scene with a blue tang fish in the bottom right corner”
Review: VAEs and Conditional VAEs

https://towardsdatascience.com/understanding-conditional-variational-autoencoders-cd62b4f57bf8
Review: Why are VAE samples blurry?

• Our reconstruction loss is the culprit
• Mean Square Error (MSE) loss looks at each pixel in isolation
• If no pixel is too far from its target value, the loss won’t be too bad
• Individual pixels look OK, but larger-scale features in the image aren’t recognizable

• Solutions?
  • Let’s choose a different reconstruction loss!
Today’s goal – learn about generative adversarial networks (GANs)

(1) Generative Adversarial Networks (GANs)

(2) Training GANs and challenges

(3) Deepfakes
Generative Adversarial Networks

(a.k.a. “GANs”)
Review: A Neural Generative Model

• Input: a point $z \in \mathbb{R}^n$ drawn from the unit normal distribution $\mathcal{N}(0, 1)$

• Output: a point $x \in \mathbb{R}^m$ distributed according to some more complex distribution

The distribution of human faces
GANs by Analogy

Scenario:
Two kids are playing a detective game (“Sherlock” or “Nancy Drew”) where one of them has to fool the other in making counterfeit dollars.
GANs by Analogy

- Initially, neither one of them is very good at their job
- The Forger produces horrible doodles on paper
- The Detective just looks for obvious “tells” / mistakes
GANs by Analogy

- As the Detective spots the Forger’s fakes, the Forger has to devise better fakes
- The Detective, in turn, has to get better at spotting the Forger’s improved fakes
GANs by Analogy

- If they keep this up long enough, the Forger gets so good that their fakes are virtually indistinguishable from the real thing...
- ...and the Detective has developed ‘superhuman’ abilities to detect them
GANs by Analogy

- GANs operationalize this idea by using neural networks to serve both of these roles.
GANs by Analogy

- GANs operationalize this idea by using neural networks to serve both of these roles.
- We call these networks the “Generator” and the “Discriminator”.
GANs: The Generator

The generator is a neural network that takes in a random vector and produces a “fake” data point.

Random Vector
(sampled from unit normal distribution)

Generator

Output of GAN trained on MNIST images
GANs: The Discriminator

The discriminator is a neural network that takes in images and predicts the probability that the image is real:

(Real image) → Discriminator → 95% probability of being real
GANs: Training the Discriminator

Discriminator wants to say:
- Real images are real with high probability.
- Fake images are real with low probability.
GANs: Training the Discriminator

Discriminator wants to maximize:

\[ E_x [\log(D(x))] + E_z [\log(1 - D(G(z)))] \]

Log probability that the real image \( x \) is predicted to be real by the discriminator.

Log probability that the fake image \( G(z) \) is predicted to be fake by the discriminator.

**Note:** Maximizing this quantity is equivalent to minimizing binary cross entropy loss with fake data labelled as 0 and real data labelled as 1.
GANs: Training the Generator

Generator wants to fool the discriminator. It wants the probability of the discriminator saying a fake image is real to be high.
GANs: Training the Generator

Generator wants to maximize:

$$E_{z}[\log(D(G(z)))]$$

Log probability that the fake image $z$ is predicted as real by the discriminator.

The generator is only allowed to change its own weights to maximize this value. Performing an update on the generator will cause all of the images to become slightly more realistic according to the discriminator.

<table>
<thead>
<tr>
<th>Random Vector</th>
<th>Before weight update</th>
<th>After weight update</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>Generator</td>
<td>(32% real)</td>
</tr>
<tr>
<td>-0.3</td>
<td>(32% real)</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>-0.3</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>-0.5</td>
<td>-0.5</td>
<td></td>
</tr>
</tbody>
</table>

**Same Random Vector**
GAN Loss

\[ E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))] \]
GAN Loss

https://jonathan-hui.medium.com/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490
GAN Loss

\[ \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D(x^{(i)}) + \log \left(1 - D(G(z^{(i)}))\right) \right] \]

What about generator?
GAN Loss

\[
\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D(x^{(i)}) + \log \left( 1 - D(G(z^{(i)})) \right) \right]
\]

\[
- \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left( 1 - D(G(z^{(i)})) \right) \quad \text{or} \quad \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left( D(G(z^{(i)})) \right)
\]

https://jonathan-hui.medium.com/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490
GAN Training Dynamics

• Does not exhibit the typical “training loss continues to go down” behavior

• Why?
  • Training a GAN is a “stalemate” – G and D continually adjust to each other’s improvements
  • More formally, training a GAN to convergence is attempting to find an equilibrium of a two-player minimax game
Demo

https://poloclub.github.io/ganlab/
What do $G$ and $D$ look like inside?

- Architecture of the networks determined by problem
What do $G$ and $D$ look like inside?

- Architecture of the networks determined by problem
- **Fully connected**

(Real image)  
\[ \rightarrow \]  
95% probability of being real

(fake image)  
\[ \rightarrow \]  
32% probability of being real
What do $G$ and $D$ look like inside?

- Architecture of the networks determined by problem
- Convolutional / Transpose convolutional

$95\%$ probability of being real

$32\%$ probability of being real
Problems with GANs
GAN training can be very unstable

- This picture? You get this if everything is working well
- Turns out, equilibria are hard to find
  - With every other net we’ve trained, the loss function is with respect to a fixed target value we’re trying to hit
  - Here, we have a “moving target” (G’s target is fool D, D’s target is detect G)
- These curves can oscillate a lot

Why do you think that is?
GAN training can be very unstable

- In particular: what happens if the discriminator ever becomes perfect at detecting G’s fakes?
  - The discriminator always returns probability zero
  - Since D is returning a constant, the gradient through D is zero
  - The generator stops training

Generator loss: $E_z[\log(D(G(z)))]$
Mode Collapse

- Generator loss says: “generate an output that looks real”
- It does not say: “generate every output that looks real”
- The generator can “cheat” by finding one output / a few outputs that reliably fool the discriminator (the specific one(s) it finds can shift over training)
Mode Collapse

Output from a healthy GAN

Output from a GAN with mode collapse. All outputs from GAN, regardless of random input noise, are the same.


How do we fix this?
Wasserstein GANs (WGANs)

\[ L_{\text{critic}}(w) = \max_{w \in W} \mathbb{E}_{x \sim \mathbb{P}_r} [f_w(x)] - \mathbb{E}_{z \sim Z} [f_w(g_\theta(z))] \]

Eq. 5: Critic Objective Function.
GAN

\[ \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D(x^{(i)}) + \log \left( 1 - D(G(z^{(i)})) \right) \right] \]

\[ -\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left( 1 - D(G(z^{(i)})) \right) \quad \text{or} \quad \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left( D(G(z^{(i)})) \right) \]

Real image \( x \)

Generator

Discriminator

\( z \sim \mathcal{N}(0, 1) \)

or

\( z \sim U\{-1, 1\} \)

\( D \rightarrow \text{cost} \)
GAN

\[
\nabla_{\theta_x} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D(x^{(i)}) + \log \left(1 - D(G(z^{(i)}))\right)\right]
\]

\[
- \nabla_{\theta_y} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D\left(G(z^{(i)})\right)\right) \text{ or } \nabla_{\theta_y} \frac{1}{m} \sum_{i=1}^{m} \log \left(D\left(G(z^{(i)})\right)\right)
\]

WGAN

\[
\text{Real image } x
\]

\[
\text{Generator}
\]

\[
\text{Critic}
\]
Any questions?
Diffusion models

• State-of-the-art models for image generation

Stable.AI  DALL·E 2

• Guest lectures by Calvin Luo (CS Ph.D. student) – Wednesday and Friday this week
Today’s goal – learn about generative adversarial networks (GANs)

(1) Generative Adversarial Networks (GANs)

(2) Training GANs and challenges

(3) Deepfakes
Deep generative models are getting really good
Is this image real or generated?
Is this image real or generated?
Is this image real or generated?
Is this image real or generated?
Is this image real or generated?
Is this image real or generated?
Is this image real or generated?
Is this image real or generated?
Is this image real or generated?
Is this image real or generated?
What is a “deep fake?”
For the purposes of this class:

• **deep·fake** \(\text{diːp feɪk}\)

• A video depicting a person in which the identity or the expression of the person’s face has been digital altered via a deep-learning-based technique.
What kinds of alterations?

- Computer vision researchers use the following scheme to talk about face appearance:

  The geometric shape and texture of a person’s face
  Which way the head is facing (rigid rotation of head; governed by neck muscles)
  Facial expression (non-rigid deformation of face; governed by facial muscles)
Two main “flavors” of deepfake

**Face swap**
- Modify identity; keep pose and expression the same
- **Application**: “digital doubles” (e.g. putting an actor’s face onto a stuntperson’s body)

**Video puppetry**
- Modify expression (+ pose); keep identity the same
- **Application**: language dubbing
Why are people worried about deepfakes?
Fake visual media has been around for a while

Fake photos

Fake videos
How deepfakes change the game

Ease/accessibility
• Now anyone with a GPU-equipped computer and some free time can create relatively convincing fakes

Scale
• A sufficiently motivated (and resourced) actor can rapidly churn out a lot of fakes
How are deepfakes made?
Two main “flavors” of deepfake

Face swap

Video puppetry

Alan Zucconi’s An Introduction to Deepfakes

Deep Video Portraits
Can deepfakes be stopped?
Detecting deepfakes

- Deep learning

  “Fighting fire with fire”

  ...but an adversary can train a model to fool your detector

[FaceForensics++]

Detecting deepfakes

- Deep learning
- “Classic” computer vision

- Find inconsistencies between movements of lips and sounds

- Compute a “fingerprint” for a person based on how facial features tend to move over time
Detecting deepfakes

• Deep learning
• “Classic” computer vision
• Social verification
Parting thoughts
“What should I do about all this?”

- **If you’re working in ML/CV research:**
  - Think critically about, and articulate, the potential real-world impacts of your work (some conferences require this now)
  - Consider contributing to detection efforts if you also work on synthesis problems

- **If you’re working on user-facing products & services:**
  - Be vigilant for fake content on your platform
  - Initiate (and sustain) serious conversations with your coworkers and employers about how to responsibly take action

- **If you’re working in the government / non-profit sector:**
  - Help educate your less-technical colleagues about how deepfakes work
  - Support (or start!) movements to draft meaningful legislation
Recap

Generative Adversarial Networks (GANs)

- Architecture
- GAN Loss + Training
- Solving problem w/ GANs → WGANs

Deepfakes

- What are deepfakes?
- Why are they a problem?
- How to detect deepfakes?