Deeksha Chauhan
HW5- Data 266

Q1. **(a) Explain the fundamental architecture of a GAN, describing the roles of the generator and discriminator networks. How do these two components interact during training, and what is the theoretical goal of this adversarial process?**

Solution:- A Generative Adversarial Network (GAN) is a deep learning architecture which is made up of two trained neural networks at the same time by means of an adversarial process:
1. Generator (G): Generates images by synthesizing noise.
2. Discriminator (D): This judges an image as being real.
(in the training set) or counterfeit (created by the generator).
These networks are in a competitive arrangement, in which:
● The generator attempts to generate believable images.
● Discriminator tries to distinguish between real and fake images.
In this adversarial system the two networks get better with each iteration.
The Process of Image generation by the Generator.
1. Latent Vector: Noise Sampling:
○ The generator begins with a random noise, which is usually denoted by z. This noise is drawn from a uniform or normal distribution.
○ The noise z is used as the starting point to be creating new, synthetic images.
2. Image Generation:
○ The noise z is then fed to the generator network that is a function, G. This capability is what the generator can use to convert the noise into an image:
■ x=G(z)x = G(z)x=G(z), x The output x is generated based on the random and is a synthetic image that looks like real images noise vector.
The training of GAN can take place in alternating phases:

1. Training the discriminator D is trained over 1 or more epochs.
2. The 1 or more epochs for training of the generator G.
3. Continue to train the D-G network using repeating steps 1 and 2.

   During discriminator (D) training process we keep Generator (G) fixed. Since discriminator training is attempting to determine how to separate real information versus fake information, it must learn to identify the imperfections of the generator. That is not a problem with a well-trained generator the same way it is with an untrained generator that generates random output.

In the same way we maintain the discriminator constant as we train the generator. Otherwise, the generator would be making attempts to strike a moving target and may converge.
This back and forth is what enables GANs to address gen. problems that otherwise can't be solved.
Goal of this adversarial process is to make the generator produce more and more realistic fake data that can be confused by a discriminator, and the discriminator continuously evolves to better differentiate between real data and the fakes of the generator.

This competition encourages both networks to reach a stable state become better until the output of the generator matches the real data distribution and forms a strong generative model.

(b) One of the major challenges in training GANs is mode collapse. Define what mode collapse is and explain why it occurs. Discuss at least two techniques that have been proposed to mitigate this problem, explaining the mechanism by which each approach helps stabilize training.

Deeksha Chauhan
HW5- Data 266

Solution:
GANs mode collapse occurs when the generator is interested in generating a few data patterns that the discriminator is fooled by. It becomes overly obsessed with a small number of dominant modes of the training data and it does not explore the entire data distribution. the generator does not explore the whole data distribution in GANs. It generates repetitive samples of data and it misses the rich variations in the actual data.
2 ways to mitigate mode collapse: -

1. Wasserstein GANs: it is an extension of traditional GANs. Here Wasserstein distance is being used a loss function instead of traditional Cross Entropy. Wasserstein Distance provide more stable and informative training signals allowing for smoother learning and reduces mode collapse. Gradient of Wasserstein distance enables better convergence. After Wasserstein distance application, Generator network outputs richness and diversity in training set.

2. Unrolled GAN:- Using a generator Loss Function which usually considers not only Discriminator's classification but also output of future discriminator versions. Which prevents (G) from over-optimizing for a single (D) and encourages it to produce more diversified and realistic samples.

(c) Describe three significant variants or extensions of the original GAN architecture (such as DCGAN, StyleGAN, Conditional GAN, CycleGAN, or others of your choice). For each variant, explain what specific problem or limitation it addresses and what innovations or modifications it introduces to achieve its goals.

Solution:-

1. **Conditional GAN: -** extends the original GAN by conditioning both the Generator and the Discriminator on additional information.
   problem it addresses : It addresses the problem of control in traditional GAN allows specific data to be generated which generates new data randomly using the characteristics that it learns through a dataset, a cGAN **adds auxiliary information,** or a condition, that the generative process follows. Standard GANs **produce outputs** based on **no guidance**, which makes them hard to control. Eg. using a GAN with a training set of handwritten digits, you can generate additional digits, yet you can't control what digit (e.g. '3' or '7') you want the model to produce, reason is that a standard version of GAN has a generator that conditions random noise into a data distribution, which does not provide any special mechanism to control the kind of output one wishes to generate.

Innovations, (G) receives both Receives both random noise and a condition (e.g., "smiling") to generate images that match the given attribute**.(D)** Takes both the image and condition as input to evaluate if the image is real and matches the condition generator learns to create images with specific features, while the discriminator learns to assess whether the image aligns with the specified conditions. This allows for fine-grained control over the generated outputs.

2. **Style GAN:-** re-architects the GAN generator to separate **"style" (semantic attributes)** from **"stochastic detail"**, enabling fine-grained, disentangled control.

Deeksha Chauhan
HW5- Data 266

Problem it addresses: Traditional GANs produce real images but with poor or less control over specific features such as hair, nose, skin tone etc. and often entangle features, causing edits to unintentionally change other traits.
key benefit of this architecture is disentangled representations in the W space leading to realistic image synthesis and editing applications.

Innovations:

- **(G) Mapping network & style space:** Transforms noise z -> intermediate code w each layer is *style-modulated* via **AdaIN**, letting coarse layers control global traits (pose/shape) and fine layers control details (color/texture).
- **(G) Per-layer noise injection:** Adds stochastic noise to each layer for natural micro-details (pores, hair strands) independent of semantic style.
- **(Regularization): Style-mixing** (apply two styles across layers) promotes disentanglement; **path-length regularization** (StyleGAN2) stabilizes scale of updates; architectural tweaks in **StyleGAN2/3** remove normalization artifacts and improve fidelity
- **(D) Discriminator:** Largely standard conv net (often with minibatch-stddev) trained adversarially to detect fakes; its stronger feedback, combined with the style mechanisms above, drives high-quality, controllable synthesis.

3. **CycleGAN** :- translate images (unpaired) to another image (e.g., horses ↔ zebras) without corresponding pairs.

Problem it addresses: The translation jobs do not have matched source target images and the supervised translation methods demand such pairs.

How it works (theory only): Uses two generators to translate in each direction as well as two discriminators to distinguish realism in the respective domains. Adds cycle consistency: forward-translation, followed by back-translation, ought to reconstruct the original image, without altering the meaning. Frequently includes identity constraint to retain colours/tones when the same already exists in the target domain.

Result: Naturally, style consistent, plausible translations of domains with only unpaired data.