

Task 3

Students:

1. Krishna Rani 10065013
2. Mohammad Sukri 10062921
3. Shilin Elizabeth Josline 10065015

1.Reparametrization trick: Explain how the trick works and what problem it addresses.

When we train the Variational Autoencoders (VAEs) model end to end, we learn the variables of the decoder and the encoder. But when the the sampling random variable is non differentiable, the decoder is part of the computation graph and gets updated but the encoder doesn't as the model breaks at the sampling random variable.

The reparametrization trick allows gradients to flow through the sampling step enabling backpropagation. Instead of sampling directly from the desired distribution, the trick is to add ϵ to z .

$$z = \mu + \sigma * \epsilon$$

This generates a number that follows a normal distribution between μ and σ . Here we get rid of the sampling operation and replaced it by a value by which we can compute the gradient.

if we use the previous sampling formula, the gradient cannot flow through, but with this, it can allowing back propagation.

2. Explain in your own words why optimisation is hard in the case of GANs. What are common ways to improve GAN training stability?

Optimisation is hard because of

1. Hard to find optima - Instead of minimizing a single objective, GANs optimize two competing objectives. The interplay between G and D can lead to oscillations or divergence instead of convergence making it hard to achieve Nash equilibrium.
2. Vanishing gradient- When the discriminator is perfect, the loss falls to 0 making the gradient vanish and thereby having no gradient to update the loss. This leaves the generator with little to no learning signal.
3. Mode collapse - Generator might collapse to a setting where it generates the same output or a limited variety of samples, ignoring large parts of the data distribution.

Common ways to improve GAN Training Stability

1. Improved Loss Functions - Use a better loss function for vanishing gradient and use an alternative loss function like Wasserstein GAN for mode collapse that avoid vanishing gradients and provide better gradient flow.

2. Label Smoothing - Replace binary labels (0 or 1) with softer targets (e.g., 0.1 and 0.9) to prevent the discriminator from becoming overconfident and overpowering the Generator
3. Gradient Penalty: Adding a gradient penalty term to the Discriminator's loss function can help prevent gradient becoming 0 and stabilize training.
4. Balanced Training - Carefully balance updates between G and D by training the generator multiple times for every discriminator update to prevent the generator from jumping from one mode to another (mode collapse problem)

3. Shortly compare GANs and VAEs. In which setting would you select one over the other?

GAN consists of two main components that compete in a minimax game.

It has a discriminator that takes data as input (either real or generated) and outputs a probability indicating whether the input is real or fake. The discriminator is trained to maximize its ability to distinguish between real and fake data.

Then the generator takes z as input and transforms it into synthetic data that resembles real data. The generator is trained to minimize the discriminator's ability to differentiate real and fake data.

A Variational Autoencoder (VAE) is end to end and doesn't have a separate "generator" (decoder). In VAE, the encoder Maps the input data x to a probability distribution in the latent space z . z is sampled from the Gaussian distribution predicted by the encoder. The decoder Maps the latent variables z back to the data space to reconstruct x . It tries to maximize evidence lower bound (ELBO).

Choose GANs when high-quality and realistic outputs are required, such as for image synthesis, style transfer, or super-resolution. Example: Generating photo-realistic faces.

Choose VAEs when you need a structured, interpretable latent space for downstream tasks, such as anomaly detection, representation learning, or semantic manipulation of generated data. Example: Detecting anomalies in medical imaging