Generative Adversarial Networks (GANs)

From Day1 why the instructor shouting for Transformation and Optimization

Why study generative modeling?

Generative models, and GANs in particular, enable machine learning to work with multi-modal outputs. For many tasks, a single input may correspond to many different correct answers, each of which is acceptable. Some traditional means of training machine learning models, such as minimizing the mean squared error between a desired output and the model's predicted output, are not able to train models that can produce multiple different correct answers.

From Day1 why the instructor shouting for transformation and Optimization

How do generative models work? How do GANs compare to others?

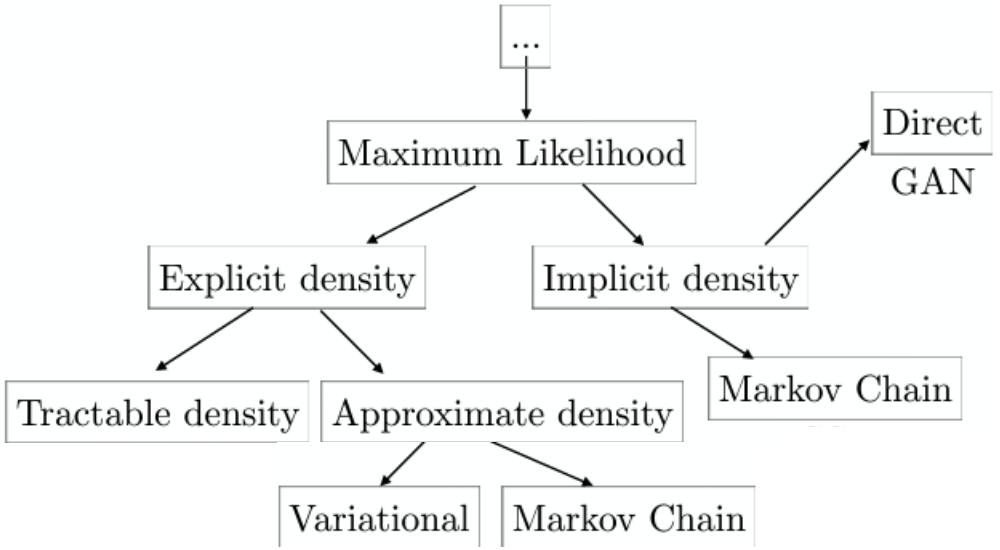
- Maximum likelihood estimation
  - To simplify the discussion somewhat, we will focus on generative models that work via the principle of maximum likelihood. Not every generative model uses maximum likelihood. Some generative models do not use maximum likelihood by default, but can be made to do so (GANs fall into this category).

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{\text{model}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right) \\ &= \arg\max_{\boldsymbol{\theta}} \log \prod_{i=1}^{m} p_{\text{model}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right) \\ &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\text{model}}\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right).\end{aligned}$$

From Day1 why the instructor shouting for transformation and Optimization

How do generative models work? How do GANs compare to others?

- Implicit density models
  - Some models can be trained without even needing to explicitly define a density functions. These models instead offer a way to train the model while interacting only indirectly with p_{model}, usually by sampling from it.



From Day1 why the instructor shouting for transformation and Optimization

How do generative models work? How do GANs compare to others?

- Tractable explicit models
  - Fully visible belief networks are models that use the chain rule of probability to decompose a probability distribution over an n-dimensional vector x into a product of one-dimensional probability distributions:

$$p_{\text{model}}(\boldsymbol{x}) = \prod_{i=1}^{n} p_{\text{model}}\left(x_i \mid x_1, \ldots, x_{i-1}\right)$$

From Day1 why the instructor shouting for transformation and Optimization

Variational Autoencoders (VAEs)

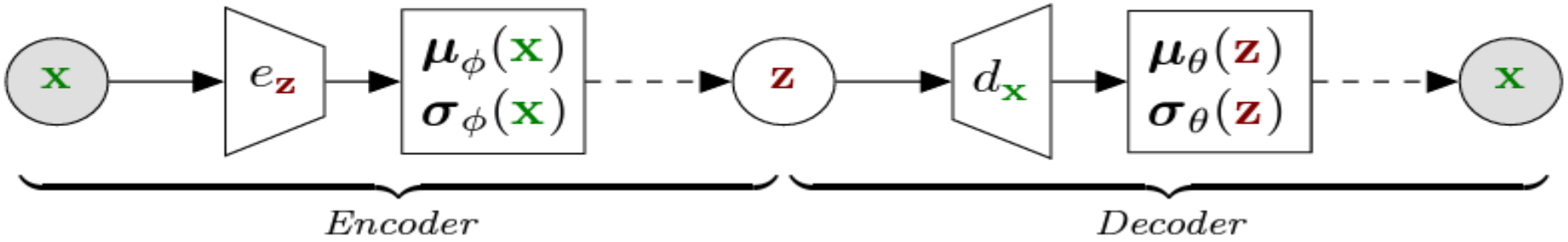
The mathematical basis of VAEs has relatively little to do with classical autoencoders (e.g. sparse autoencoders or denoising autoencoders). As mentioned earlier, our aim is to maximize the probability of each X in the training set under the entire generative process, according to:

$$P(X) = \int P(X|z;\theta)P(z)dz$$

From Day1 why the instructor shouting for transformation and Optimization

Variational Autoencoders (VAEs)

N (·;μ,Σ) denotes a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ, diag{·} is the operator that forms a diagonal matrix from a vector by putting the vector entries on the diagonal, 0_L is the zero-vector of size L, and I_L is the identity matrix of size L. $p_{\theta x}(x)$ is a generic notation for a parametric pdf of the random variable x, where θx is the set of parameters. It is equivalent to p(x;θx).



For simplicity of presentation and consistency across models, in the present review, $p_{\theta x}(x|z)$ is assumed to be a Gaussian distribution with diagonal covariance matrix for all models; that is,

$$p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\theta_{\mathbf{x}}}(\mathbf{z}), \mathrm{diag}\{\boldsymbol{\sigma}^2_{\theta_{\mathbf{x}}}(\mathbf{z})\})$$
$$= \prod_{f=1}^{F} p_{\theta_{\mathbf{x}}}(x_f|\mathbf{z}) = \prod_{f=1}^{F} \mathcal{N}(x_f; \mu_{\theta_{\mathbf{x}},f}(\mathbf{z}), \sigma^2_{\theta_{\mathbf{x}},f}(\mathbf{z})).$$

From Day1 why the instructor shouting for transformation and Optimization

Denoising auto-encoder (DAE)

The Denoising Auto-Encoder (DAE) introduces a crucial modification to the traditional auto-encoder (AE) framework by incorporating a corruption process during training. This modification aims to enhance the robustness and generalization capabilities of the model by forcing it to learn from noisy or corrupted input data. The fundamental concept behind DAE lies in its ability to reconstruct the original clean input data $\boldsymbol{x}$ from its corrupted counterpart $\hat{x}$. The corruption process, represented by M_D, generates noisy versions of the input data, simulating real-world scenarios where the data may be corrupted by noise or other sources of interference.

$$\tilde{\mathbf{x}}^{(i)} \sim \mathcal{M}_D\left(\tilde{\mathbf{x}}^{(i)} \mid \mathbf{x}^{(i)}\right)$$

$$L_{\mathrm{DAE}}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{x}^{(i)} - f_\theta\left(g_\phi\left(\tilde{\mathbf{x}}^{(i)}\right)\right)\right)^2$$

From Day1 why the instructor shouting for transformation and Optimization

Classical Auto-Encoder Models

Auto-encoders were originally proposed as artificial neural networks designed for dimension reduction. The fundamental concept behind classical auto-encoders involves constructing an architecture with input and output layers of the same dimension as the data while introducing intermediate layers of smaller dimensions to create a bottleneck. This bottleneck forces the network to learn a compressed representation of the input data, capturing its essential features. Classical auto-encoders have evolved over time to address various challenges and limitations inherent in the original design. One significant advancement of classical Auto-Encoders is the introduction of Denoising Auto-Encoders (DAE). DAEs are designed to reconstruct clean data from corrupted inputs, making them robust to noise and enhancing their generalization capabilities. Another notable development is the Stacked Convolutional Auto-Encoders (scAE). scAEs leverage convolutional layers to capture spatial dependencies and hierarchical features in the input data, making them particularly effective for imagerelated tasks. Additionally, Robust Deep Auto-Encoders (RDA), aim to enhance the robustness of auto-encoder models to outliers and anomalies in the input data.

From Day1 why the instructor shouting for transformation and Optimization

Diffusion models

Diffusion models have emerged as the state-of-the-art generation paradigm for a wide range of vision generation tasks. As recent challengers, autoregressive models share several characteristics with diffusion models. The methods are capable of generating diverse, high-quality samples, yet both also suffer from inefficient inference due to their iterative or sequential generation processes. Additionally, both approaches are likelihood-based and optimize objectives such as Negative Log-Likelihood (NLL) or Evidence Lower Bound (ELBO), making them relatively easy to train. Recent advances have focused on compressing visual content into latent spaces to improve the efficiency of high-resolution generation. Diffusion models employ a predefined forward process that gradually corrupts data and an iterative denoising process to recover samples. This iterative process retains spatial locality at each step, which benefits visual coherence. However, the necessity to corrupt the training data with Gaussian noise may potentially limit their scalability for understanding tasks. Additionally, while diffusion models operate primarily in continuous spaces, their discrete variants still lag behind in performance, posing challenges for multimodal applications like text generation. Autoregressive models, by contrast, inherently introduce unidirectional biases and discretization, which might not be ideal for visual tasks, partially explaining their lag behind diffusion models in recent benchmarks. However, autoregressive models offer flexibility in handling diverse modalities and combining generation with understanding. This adaptability aligns with the emerging trend of integrating autoregressive models with Large Language Models (LLMs) to create a unified framework for multimodal input-output tasks, bridging both generation and understanding capabilities.

From Day1 why the instructor shouting for transformation and Optimization

Masked Autoencoder.

Masked autoencoders (MAEs) are designed to learn robust data representations by randomly masking portions of the input data and training the model to reconstruct the missing content. Both MAEs and autoregressive models share similarities as they compress images into discrete sequences of visual elements and model these sequences. In fact, MAEs and autoregressive models inherit two leading paradigms from natural language processing (NLP): MAEs adopt the BERT-style encoder-decoder structure, while autoregressive models follow the GPT-Style decoder-only approach. Nevertheless, there are essential differences between them:

1) MAEs are trained by randomly masking visual tokens and reconstructing the missing part, while autoregressive models are trained to predict the next element in an ordered sequence;
2) MAEs utilize full attention, enabling each token to consider the entire surrounding contexts, whereas autoregressive models typically employ causal attention, restricting each token to focus only on previous ones;
3) During decoding, MAEs randomly generate multiple tokens in parallel, while autoregressive models generate tokens sequentially in a predefined order. These differences lead to distinct strengths: MAEs generally excel in representation learning and visual understanding tasks, and recent works like MaskGIT and MAGE also extend the Masked Image Modeling (MIM) approach to image generation.

From Day1 why the instructor shouting for transformation and Optimization