Anup Tuladhar
May 31, 2019

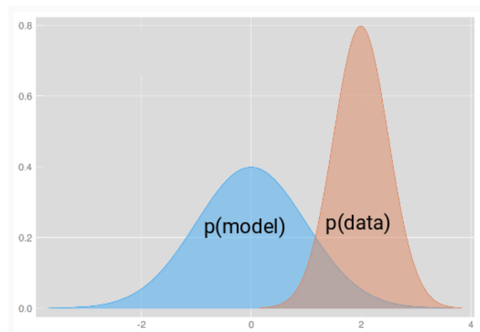
Generative Models

Generative models are a set of unsupervised learning approaches to estimate the underlying distribution that produces our observed data in order to generate new data points that are likely to come from the same distribution.

- Let x = data (e.g. features, image) and y = target (e.g. label)
- Generative models estimate the joint probability of p(x, y)
- Contrast to Discriminative models, which model conditional probability p(y | x)

In general, we want to minimize the KL divergence between our data distribution and generated distribution

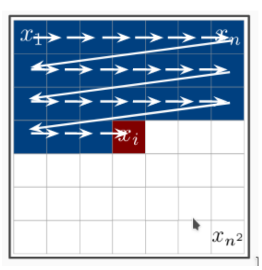
$$\min_{p \in \mathcal{P}_{\mathbf{x},\mathbf{z}}} D_{\mathrm{KL}}\left(p_{\mathrm{data}}(\mathbf{x}) \,\|\, p(\mathbf{x})\right)$$



Generative models can be broadly split up into 3 types:

1. Models that explicitly estimate the probability density and generate samples from this explicity estimate (e.g. autoregressive models like WaveNet, PixelRNN).
2. Models that approximate the probability density and generate samples from this approximation (e.g. variational autoencoders).
3. Models that implicitly estimate the probability density function. In other words, they generate samples without trying to estimate the probability density function (e.g. generative adversarial networks)

Autoregressive models (Explicit density estimation)
Treating data like a time-series (e.g. treating an image like a succession of pixels), they make predictions of the "next pixel" based on all previous pixels.



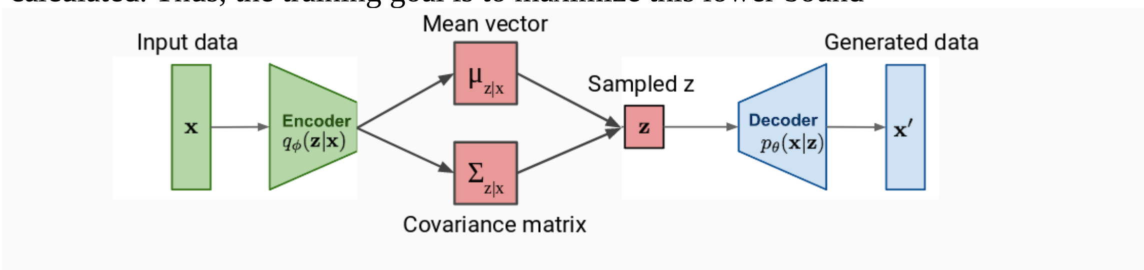
$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i | x_1, x_2, \ldots, x_{i-1}) = \prod_{i=1}^{n} p(x_i | \mathbf{x}_{<i})$$

Likelihood of data (e.g. an image) is the product of conditional probabilities of all segments (e.g. pixels)

Training goal is to maximize the likelihood

Variational autoencoders (Approximate density estimation)

- Use an autoencoder to “compress” the data into a distribution in latent space
  - Often this latent space is assumed to come from a Gaussian distribution with a mean of 0 and a covariate matrix equal to the identity matrix
- The latent space can be sampled from to generate new samples
- The likelihood of p(x) is not fully solveable, but a variational lower bound (ELBO) can be calculated. Thus, the training goal is to maximize this lower bound

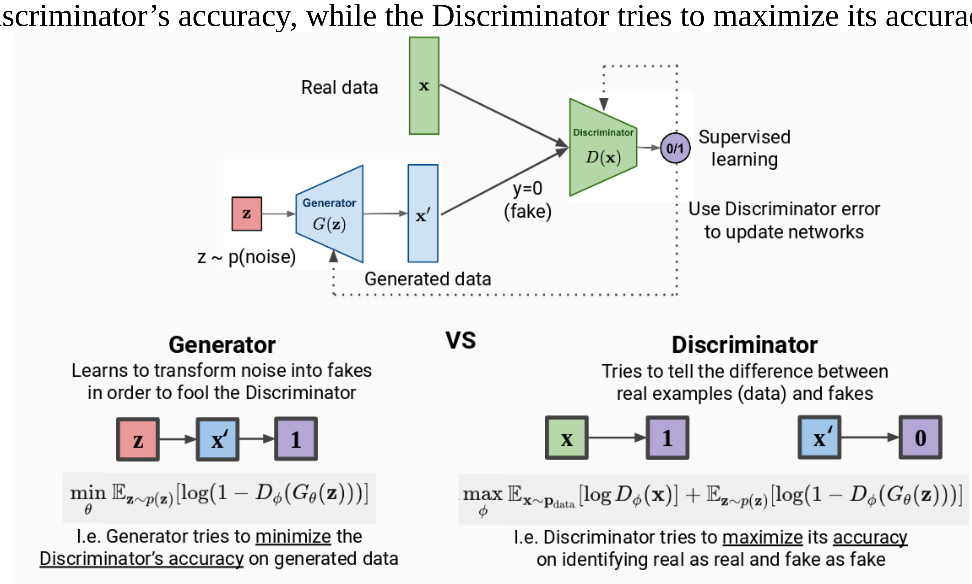


$$\log p_\theta(x^{(i)}) = \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\text{Solvable}} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\text{Intractable, but know it is} \geq 0}$$

Variational Lower Bound (ELBO) $= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))$

Generative Adversarial Networks (Implicit density estimation)

- Use a game-theory approach to train a generator that can produce realistic samples
- The generator is pitted against a discriminator whose explicit goal is to tell the difference between real examples and fake examples
- The theoretical objective function is a min-max game, where the Generator tries to minimize the Discriminator’s accuracy, while the Discriminator tries to maximize its accuracy



GAN objective function: $\min_\theta \max_\phi V(G_\theta, D_\phi) = \mathbb{E}_{\mathbf{x}\sim \mathbf{p}_{\text{data}}}[\log D_\phi(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D_\phi(G_\theta(\mathbf{z})))]$

- For practical reasons, in training the generator uses an alternate objective function, which aims to maximize the Discriminator’s error instead of minimize its accuracy

$$\max_\phi \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(D_\phi(G_\theta(\mathbf{z})))]$$