DS 4440

Auto-encoders & Diffusion Models

Many problems in ML amount to generative modeling.

Consider image generation

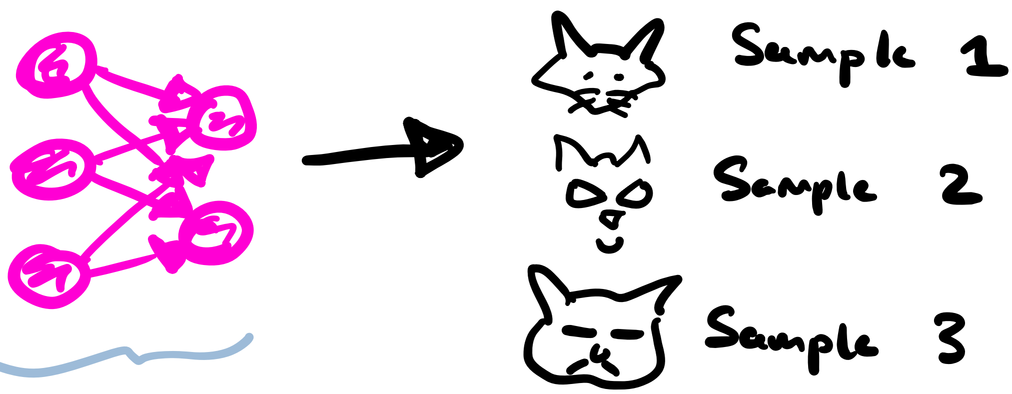


Image generation model

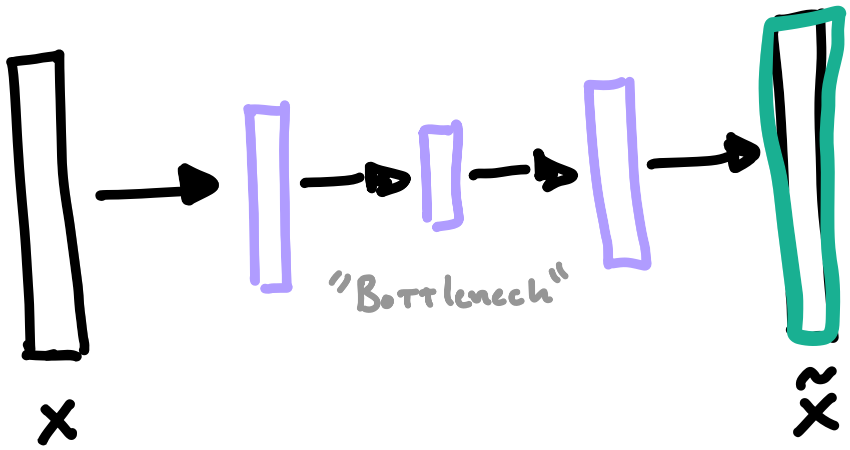
Hopefully a bit better though

Want to be able to sample from some (unknown) distribution

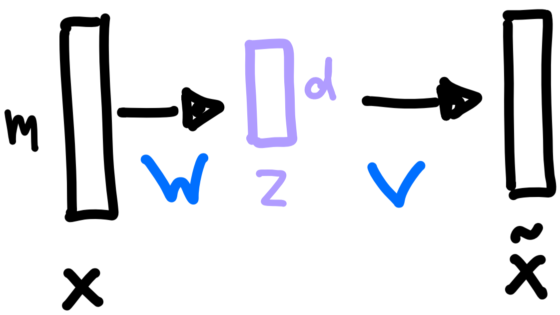
$p^*(x)$ cat image distribution

But this hard. (What would P look like for natural images?)

Auto-encoders!



Consume x, compress it, then reconstruct it.



$$\underset{}{z} = \underbrace{W}_{(d\times m)}\underbrace{x}_{(m\times 1)} \qquad \text{Encoder}$$

$$\tilde{x} = \underbrace{V}_{(m\times d)}\underbrace{z}_{(d\times 1)} \qquad \text{Decoder}$$

Loss: Reconstruction error

$$\mathcal{L}(x, \tilde{x}) = MSE(x, \tilde{x})$$

De-noising Auto-encoders add noise first

$$z = Wx' \qquad x' = x + \epsilon$$

$$\epsilon \sim N(\vec{0}, \sigma^2)$$

Interpolating in latent space

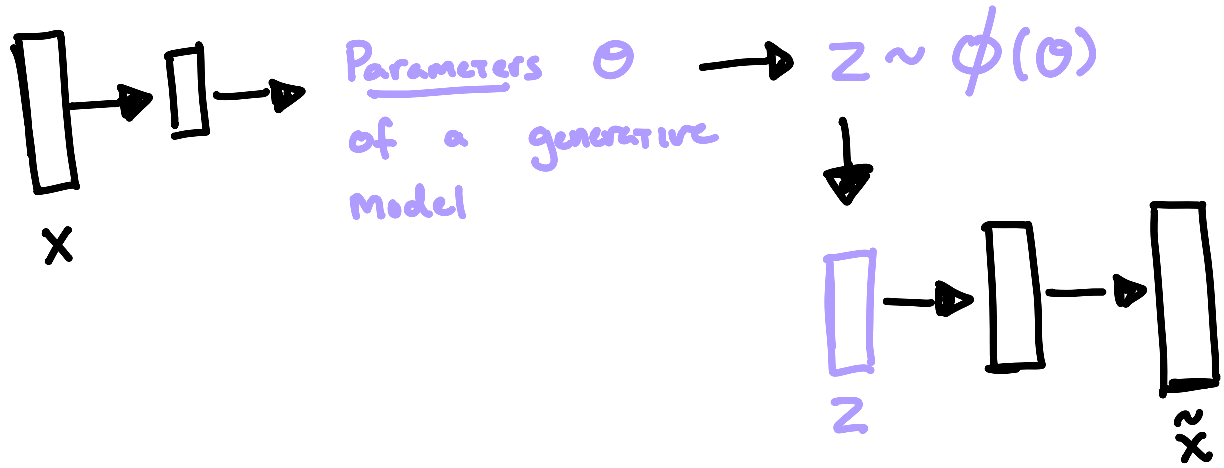
$$h_1 \leftarrow enc(x_1)$$

$$h_2 \leftarrow enc(x_2)$$

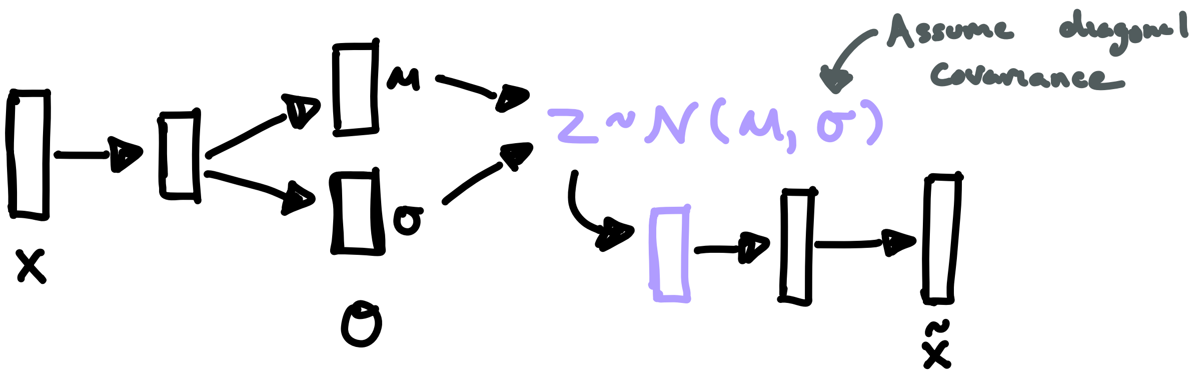
$$h_{1,2}^{\lambda} \leftarrow \lambda h_1 + (1-\lambda) h_2$$

$$\tilde{x}_{1,2} \leftarrow dec(h_{1,2}^{\lambda})$$

Variational Auto-encoders inject a probabalistic element into the process.



In practice: $\phi = N(\underbrace{\mu, \sigma}_{\theta})$



For a Normal Distribution, this is equivalent to:

$$z = \mu + \sigma \cdot N(\phi, I)$$

$$= \underbrace{\mu + \sigma}_{\text{Torch Params.}} \cdot \epsilon$$

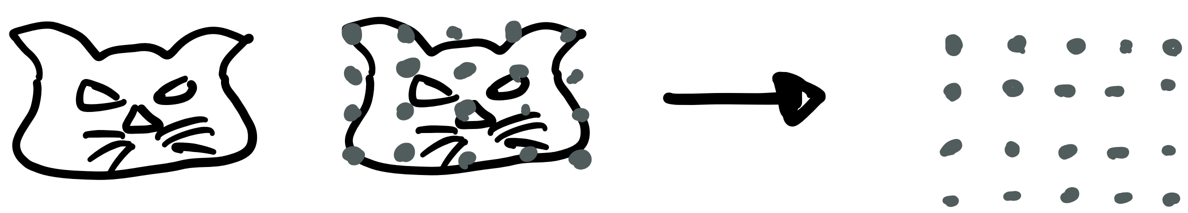
[See CoLab]

Diffusion Models approximate P by iteratively sampling from simple distributions.

Suppose we have a sample cat image x_0. In **Gaussian Diffusion** we repeatedly noise this

$$x_{t+1} \leftarrow x_t + \eta_t$$

$$\eta_t \sim N(0, \sigma^2)$$



1 2 T

Now consider de-noising iteratively: Given P_t, produce distribution P_{t-1}. This is a **Reverse Sampler**.

IDEA Learn to reverse noising one step at a time.

$$P(x_{T-1} \mid \underbrace{x_T = z}_{\text{Sample from } P_T})$$

But how? For Gaussian noising it turns out (for small σ)

$$P(x_{T-1} \mid x_T = z) \approx \mathcal{N}(x_{T-1} \mid \mu_T, \sigma^2)$$

So we have

$$\boxed{x_0} \quad x_1 \quad x_2 \quad \cdots \quad x_T$$

Actual image

noised drawn from

$$P_1 \quad P_2 \quad \cdots \quad P_T$$

We want a Reverse Sampler to yield $\mu_T(z) \doteq \mathbb{E}[x_T \mid \underbrace{x_{T+\Delta T} = z}_{\text{Sample } \Delta T \text{ steps away}}]$

Estimate via learned $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$

Learn via denoising objective

$$\min \mathbb{E}_{\underbrace{x_\tau, x_{\tau+\Delta\tau}}_{\text{Because we are sampling}}} \underbrace{\| f_\theta(x_\tau + \Delta_\tau) - x_\tau \|_2^2}_{\text{Learn to denoise a sample}}$$

If we learn this, we can perform one denoising step.

But then we can generate!

$\hat{x}_T \sim N(0, \sigma^2)$ Sample pure noise

For _ in $\{T-1 \dots 0\}$

$\hat{x}_t \sim N(f_\theta(\underbrace{\hat{x}_\tau}_{\text{Prior sample}} + \Delta_\tau), \sigma^2)$

Return $\hat{x}_T$

How could we condition the generation?