TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2022

Diffusion Model Basics

Denoising Diffusion Probabilistic Models

Ho, Jain and Abbeel, June 2020

Progressive VAEs

Diffusion models are a special case of progressive VAEs.

A progressive VAE has layers of latent variables $z_1, \ldots, z_L$.

The encoder defines $P_{\mathrm{enc}_0}(z_1|y)$ and $P_{\mathrm{enc}_\ell}(z_{\ell+1}|z_\ell)$ (defining a Markov chain).

We have a prior $P_{\mathrm{pri}}(z_L)$ and a decoder defined by $P_{\mathrm{dec}_\ell}(z_\ell|z_{\ell+1})$ and $P_{\mathrm{dec}_0}(y|z_1)$.

Ho et al. take $L = 1000$.

The Ho et al. Diffusion Model

The encoder is not trained — The encoder just adds noise.

The encoder is designed so that Z_L is distributed as $\mathcal{N}(0, I)$. The prior is not trained.

The decoder is a single network applied to all layers but taking the layer as an argument.

A Gaussian Noise Encoder

We assume $y \in R^d$ and $z_\ell \in R^d$.

In the case of images every z_ℓ is an image with the same dimension as y.

For notational convenience we define $z_0 = y$.

The index ℓ will always range from 1 to L where $z_{\ell-1}$ might be z_0.

The encoder is fixed (not trained) and is defined by a sequence of noise levels $\sigma_1, \ldots, \sigma_L$ where for $1 \leq \ell \leq L$ we have

$$z_\ell = \sqrt{1 - \sigma_\ell^2}\, z_{\ell-1} + \sigma_\ell \epsilon \quad \epsilon \sim \mathcal{N}(0, I)$$

A Gaussian Noise Encoder

$$z_\ell = \sqrt{1-\sigma_\ell^2}\, z_{\ell-1} + \sigma_\ell \epsilon \qquad \epsilon \sim \mathcal{N}(0, I)$$

This is designed so that if z_ℓ as unit variance in each dimension then $z_{\ell+1}$ also has unit variance in each dimension.

$z_0 = y$ is scaled so that each coordinate is in the interval $[0, 1]$.

The constant variance of z_ℓ is important because the same decoder is being used at all ℓ and we want the scale of the decoder input to be independent of ℓ.

A Gaussian Noise Encoder

$$z_\ell = \sqrt{1 - \sigma_\ell^2}\, z_{\ell-1} + \sigma_\ell \epsilon \quad \epsilon \sim \mathcal{N}(0, I)$$

Note that the mean of z_ℓ equals $\sqrt{1 - \sigma_\ell}$ times the mean of $z_{\ell-1}$.

This repeated reduction drives the mean of z_L to a negligible level.

We then get that z_L is distributed as $\mathcal{N}(0, I)$.

A Gaussian Noise Encoder

$$z_\ell = \sqrt{1 - \sigma_\ell^2}\, z_{\ell-1} + \sigma_\ell \epsilon \quad\quad \epsilon \sim \mathcal{N}(0, I)$$

They increase the noise at higher levels. After some experimentation they use

$$\sigma_\ell^2 = 10^{-4} + .02\,\frac{\ell}{L} \quad \text{(with } L = 1000\text{)}$$

Direct Sampling of z_ℓ

We can sample from $P_{\mathrm{enc}}(z_\ell|z_0)$ directly

$$\text{define} \quad \alpha_\ell = \prod_{\ell=1}^{\ell} \sqrt{1-\sigma_\ell^2}$$

$$z_\ell = \alpha_\ell z_0 + \sqrt{1-\alpha_\ell^2}\,\epsilon \quad \epsilon \sim \mathcal{N}(0, I)$$

The variance of the noise term can be derived by solving a recurrence relation or by observing that z_ℓ must have unit variance for unit variance z_0.

A Natural but Problematic Loss Function

$$\text{dec}^* = \underset{\text{dec}}{\text{argmin}} \;\; E_{z_0,\ell,z_{\ell-1},z_\ell} \;\; ||z_{\ell-1} - \text{dec}(z_\ell, \ell)||^2$$

Here $z_{\ell-1}$ is sampled directly from z_0 and z_ℓ is sampled from $z_{\ell-1}$.

Reducing the Decoder's Dependence on ℓ.

We have already reduced the decoder's dependence on ℓ by making z_ℓ have unit variance for all ℓ.

But predicting $z_{\ell-1}$ from z_ℓ behaves differently for different ℓ.

For small ℓ, where σ_ℓ is small, $z_{\ell-1}$ is near z_ℓ. For large ℓ we have that $z_{\ell-1}$ is far from z_ℓ. Since the same network is used at all ℓ we want to reduce the dependence on ℓ.

This can be accomplished by using an ϵ-decoder.

An ϵ-Decoder

First we solve for $z_{\ell-1}$ in terms of z_ℓ and ϵ.

$$z_\ell = \sqrt{1-\sigma_\ell^2}\, z_{\ell-1} + \sigma_\ell \epsilon \quad\quad \epsilon \sim \mathcal{N}(0, I)$$

$$z_{\ell-1} = \frac{1}{\sqrt{1-\sigma_\ell^2}}(z_\ell - \sigma_\ell \epsilon)$$

$$\mathrm{dec}(z_\ell, \ell) = \frac{1}{\sqrt{1-\sigma_\ell^2}}\,(z_\ell - \sigma_\ell\, \epsilon_\Phi(z_\ell, \ell)) + \sigma_\ell \delta \quad\quad \delta \sim \mathcal{N}(0, I)$$

Here $\epsilon_\Phi(z_\ell, \ell)$ is a trained network whose target value has the same behavior at all levels of ℓ.

An ϵ-Decoder

$$\mathrm{dec}(z_\ell, \ell) = \frac{1}{\sqrt{1-\sigma_\ell^2}}\left(z_\ell - \sigma_\ell\, \epsilon_\Phi(z_\ell, \ell)\right) + \sigma_\ell \delta \qquad \delta \sim \mathcal{N}(0, I)$$

However, SGD on the loss $||z_{\ell-1} - \mathrm{dec}(z_\ell, \ell)||^2$ now scales the SGD gradients on ϵ differently for different ℓ.

We effectively have different learning rates for different ℓ.

Training the ϵ-Decoder

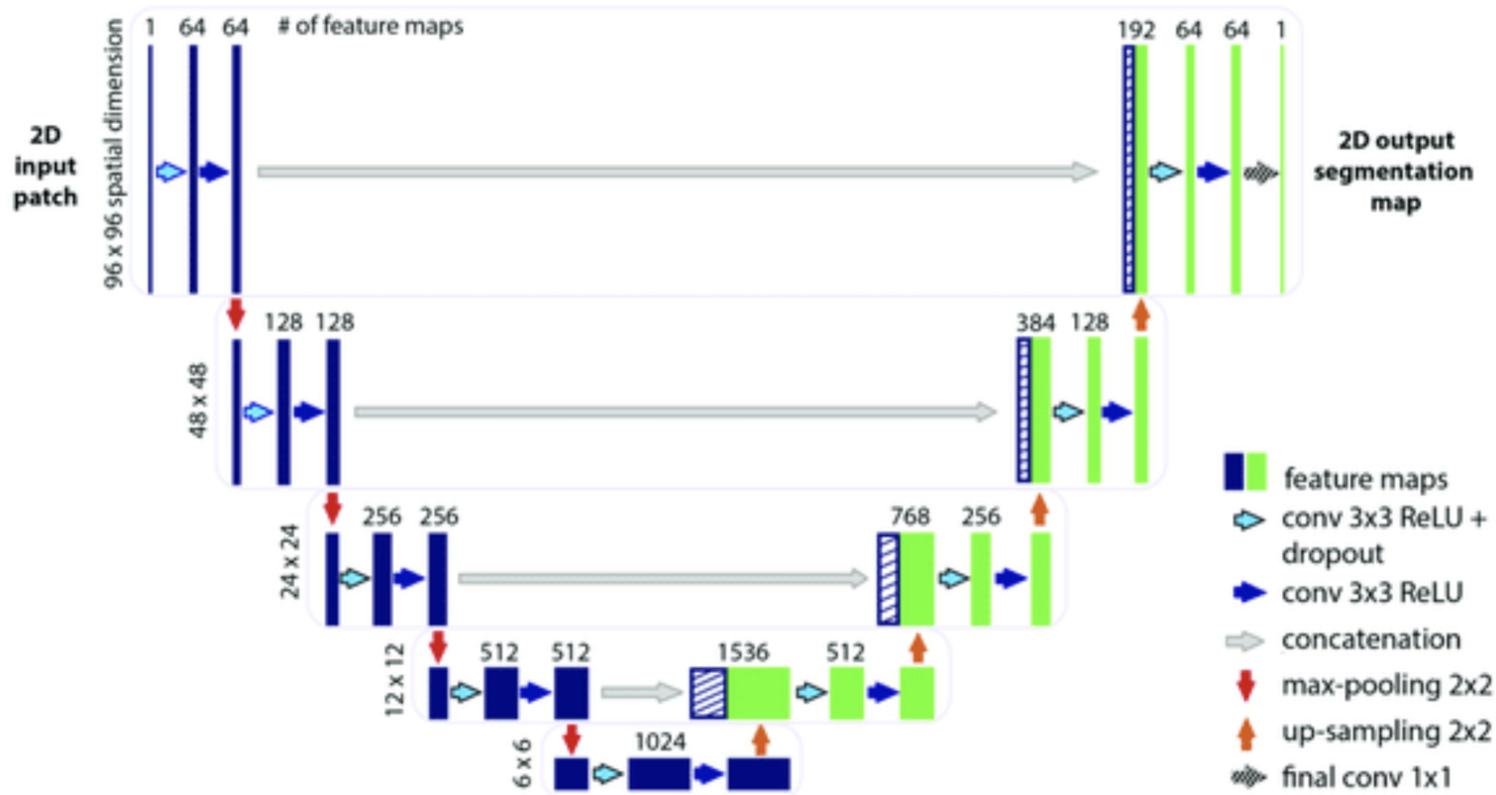
To make the scale of the SGD gradients independent of ℓ we use the following loss.

$$\epsilon^* = \underset{\epsilon}{\mathrm{argmin}} \left\{ \begin{array}{l} E_{z_0,\ell,z_{\ell-1},\epsilon\sim\mathcal{N}(0,I)} \\ \\ \quad\quad ||\epsilon - \epsilon_\Phi\left(z_\ell(z_{\ell-1},\epsilon),\ell\right)||^2 \end{array} \right.$$

We now repeatedly sample z_0, ℓ, $z_{\ell-1}$ and ϵ and do gradient updates on ϵ_Φ.

ϵ-Decoder Architecture

The ϵ-decoder is a U-Net.



Voila

But this is “mearly” a face generator. DALLE and DALLE-2 do text-conditioned image generation.

END