Diffusion Models (ch. 20 Bishop)

Wednesday, November 13, 2024 7:12 AM

§20 Diffusion Models (still following Bishop's book)

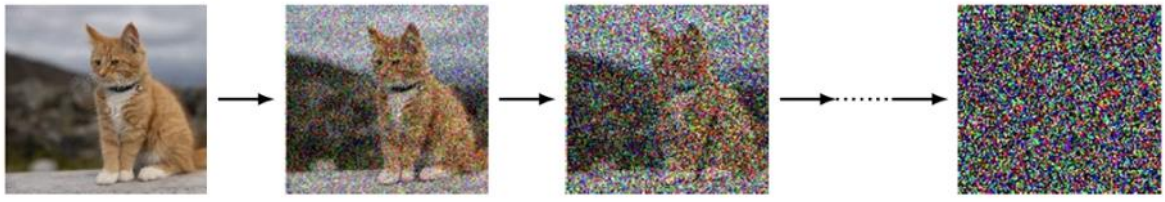
aka "Denoising diffusion probabalistic models" or DDPM — Sohl-Dickstein et al. '15, Ho, Jain, Abbeel '20

Other generative approaches: GANS, VAE, Normalizing Flow

Diffusion models: easier to train, but slow to generate new samples

State-of-the-art!

Inspiration: we corrupt training data



$z_1 = \sqrt{1-\beta_1}\, x + \sqrt{\beta_1}\, \varepsilon_1 \quad \varepsilon_1 \sim N(0, I)$

$\vdots$

$z_t = \sqrt{1-\beta_t}\, z_{t-1} + \sqrt{\beta_t}\, \varepsilon_t$

pushes mean toward 0, variance toward I

In lit., "x_0" is our "x", "x_t" is our "z_t"

$0 < \beta_1 < \beta_2 < \ldots < \beta_T < 1$ chosen by hand

z_{t-1} —$q(z_t|z_{t-1})$→ z_t, ←$q(z_{t-1}|z_t, x)$

(Fixed) $q(z_t|z_{t-1}) = N(\sqrt{1-\beta_t}\, z_{t-1}, \beta_t I)$

"Forward Process" (like "encoder" in VAE, but not learned)

Altogether, (z_t) is a Markov Chain. Conditioned on starting point (observed data x),

$$q(z_1, \ldots, z_T | x) = q(z_1|x) \cdot \prod_{t=2}^{T} q(z_t|z_{t-1})$$

So ("marginalize" over $z_1, \ldots, z_{T-1}$)

$$q(z_T|x) \sim N(\sqrt{\alpha_T}\, x, (1-\alpha_T) I)$$

"diffusion kernel"

$\alpha_T = \prod_{t=1}^{T} (1-\beta_t) \to 0$ as $T \to \infty$ (since $0 < \beta_t < 1$)

(can do this for any T. Useful for training as we can jump to any T w/o starting at beginning)

Goal: learn to undo this forward process

Then we can sample z_∞ (i.e. z_T for T large) from $N(0, I)$, go backward, and end w/ a sample $x \sim p(x)$

$$q(z_{t-1}|z_t) = \frac{q(z_t|z_{t-1}) \cdot q(z_{t-1})}{q(z_t)}$$

($q(z_t|z_{t-1})$: known; $q(z_{t-1})$: intractable: $q(z_{t-1}) = \int q(z_{t-1}|x) p(x)\, dx$, unknown; $q(z_t)$: irrelevant for now)

So instead look at

$$q(z_{t-1}|z_t, x) = \frac{q(z_t|z_{t-1}, \cancel{x}) \cdot q(z_{t-1}|x)}{q(z_t|x)}$$

($q(z_t|z_{t-1})$: diffusion kernel; x: irrelevant due to Markov structure)

For training, $|x$ is ok

For inference, it isn't!

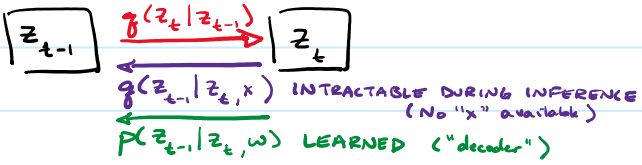
$$= N(m_t, \sigma_t^2 I) \text{ w/ } m_t = \frac{(1-\alpha_{t-1})\sqrt{1-\beta_t}\, z_t + \sqrt{1-\alpha_{t-1}}\, \beta_t\, x_t}{1-\alpha_t}$$

$$\sigma_t^2 = \frac{\beta_t (1-\alpha_{t-1})}{1-\alpha_t}$$

Diffusion Models (p. 2)

Wednesday, November 13, 2024 9:52 AM

Reverse Decoder we'll learn model parameters w to approximate $q(z_{t-1}|z_t)$ the reverse distribution.



Keep $\beta_t \ll 1$ so forward encoding is small (almost identity)

so easier to invert.

Then use many steps ($T \approx 1000$'s) so $\alpha_T \approx 0$

We'll parameterize

$$p(z_{t-1}|z_t, w) = N(\mu, \beta_t I)$$

often (not always) fixed.

$\mu(z_t, w, t)$ via neural net.

overall,

$$p(x, z_1, \dots, z_T | w) = p(z_T) \cdot \left(\prod_{t=2}^{T} p(z_{t-1}|z_t, w) \right) \cdot p(x|z_1, w)$$

$p(z_T)$: $\sim N(0, I)$ seed; $p(z_{t-1}|z_t, w)$: neural net eval

Conceptually straightforward (but $T \approx 1000$'s makes it slow)

Training

Objective is likelihood, $p(x|w) = \int \dots \int p(x, z_1, \dots, z_T | w)\, dz_1 \dots dz_T$

$\int p(\vec{z})$ intractable

via neural net

Intractable

Overcome (approximate) w/ standard trick

("the" trick of "Variational Inference", used in VAE and EM) see also §16.3

"Evidence Lower BOund" (ELBO)

Def: The Kullback-Leibler (KL) divergence of two pdfs f, g

$$KL(f \| g) = -\int f(z) \cdot \ln\left(\frac{g(z)}{f(z)}\right)$$

see §2.5.7 §2.5.5

⚠ Not symmetric in f, g

Thm $KL(f \| g) \geq 0$ (and $= 0$ iff $f = g$ a.e.)

proof sketch: Use convexity of $-\log(\cdot)$ w/ Jensen's inequality.

Then

$$\ln(p(x|w)) = \underbrace{\int q(\vec{z}) \cdot \ln\left(\frac{p(x, \vec{z}|w)}{q(\vec{z})}\right) d\vec{z}}_{\mathcal{L}(w)} - \underbrace{\int q(\vec{z}) \cdot \ln\left(\frac{p(\vec{z}|x, w)}{q(\vec{z})}\right) d\vec{z}}_{= KL(q,\, p(\vec{z}|x,w)) \geq 0}$$

$= p(\vec{z}|x,w) \cdot p(x|w)$; IGNORE

want to maximize this...

"$\vec{z}$" really means $z_1, \dots, z_T$

$\int q(\vec{z})$ tractable

"evidence lower bound" tractable since we know q (see Eq. 20.26, use M.C.)

$\geq \mathcal{L}(w)$

... instead, maximize this (lower bound.

$= 0$ if $q(\vec{z}) = p(\vec{z}|x,w)$

(we choose q to make tractable, but also attempt

Diffusion Models (p. 3)

Wednesday, November 13, 2024 10:36 AM

(improvement: predicting noise, §20.2.4) Ho, Jain, Abbeel '20

... plus simplifications, approximations ...

Ultimately,

$$\text{loss}(\omega) = -\sum_{t=1}^{T} \| g(\sqrt{\alpha_t}\, x + \sqrt{1-\alpha_t}\, \varepsilon_t, \omega, t) - \varepsilon_t \|^2$$

predicted noise; inputs; neural net; training data; weights; (per training sample)

in practice, subsample (SGD style) during training

Other training speedups to relax Markovian structure, speed up training (Song, Meng, Ermon '20)

Inference: generating new samples $x \sim p(x)$

$z_T \sim N(0, I)$

for $t = T, T-1, \ldots, 2$

$$\alpha_t = \prod_{\tau=1}^{t} (1-\beta_\tau)$$

$$\mu(z_t, \omega, t) \leftarrow \frac{1}{\sqrt{1-\beta_t}} \left(z_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\, g(z_t, \omega, t) \right)$$ (neural net)

$\varepsilon \sim N(0, I)$

$z_{t-1} \leftarrow \mu(z_t, \omega, t) + \sqrt{\beta_t} \cdot \varepsilon$

$$x = \frac{1}{\sqrt{1-\beta_1}} \left(z_1 - \frac{\beta_1}{\sqrt{1-\alpha_1}}\, g(z_1, \omega, t) \right)$$

Slow since $T \approx 1000$!

Improvement: view as solving stochastic diff. eq. and use better discretization

Relation to score-matching (Hyvärinen '05, Song, Ermon '19)

Stein score $s(x) = \nabla_x \ln p(x)$ (ignores multiplicative normalization)

loss $J(\omega) = \frac{1}{2} \int \| s(x, \omega) - \nabla_x \ln p(x) \|^2 p(x) dx$

$s(x, \omega)$: approximate score fcn via neural net