Sampling (ch. 14 from Bishop's "Deep Learning")

Tuesday, November 12, 2024 2:18 PM

Last part of our class will focus on generative models and I plan to follow Christopher Bishop's "Deep Learning" book from 2024

- ch 14 sampling (rejection, MCMC, Langevin) background
- ch 17 GANs
- ch 18 Normalizing Flows
- ch 19 (variational) Autoencoders (VAE)
- ch 20 diffusion models

GOAL: sample $z \sim p(z)$ ("distribution" / PDF...)

Ch 14: background on sampling

Given: a source that samples $z \sim \text{Unif}(0,1)$ or $z \sim N(0,1)$

- Inverse transform sampling for simple distributions (details: see §14.1.2 or wikipedia)

Ex: you can sample $z \sim \text{Unif}(0,1)$] source

you want samples from $\text{Unif}(-1,1)$] target

Solution: $y = g(z)$, $g(z) := 2 \cdot z - 1$

- Rejection sampling

motivation: ① p is a function we know and in fact our end goal is to find Z_p!

② $p(z|y) = \frac{p(y|z)p(z)}{p(y)}$ ← unknown

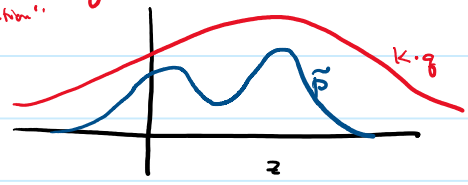
Want to sample from pdf $p(z)$

Assume we can sample from an unnormalized version $\tilde{p}$

i.e. $p(z) = \frac{1}{Z_p} \cdot \tilde{p}(z)$ $\quad Z_p = \int \tilde{p}$ = "partition function"

Use a simple ("proposal distribution" or "envelope function") q s.t. $\exists\ K > 0$ (i.e. one we already know how to sample from)

$\tilde{p}(z) \le K \cdot q(z)$ (comparison function)



To generate a sample:

Until "ACCEPT"

1) draw $z \sim q(z)$

2) with probability $\frac{\tilde{p}(z)}{K \cdot q(z)}$, ACCEPT (else: REJECT)

i.e. draw $u_0 \in \text{Unif}[0, K \cdot q(z)]$ and accept if $u_0 \le \tilde{p}(z)$

Inefficient if it takes a while to accept... want $\tilde{p} \approx K \cdot q$

Variants: adaptively refine q

Sampling (p. 2)

Tuesday, November 12, 2024 2:42 PM

- **Importance Sampling** For related problem of estimating

$$\mathbb{E}_{z\sim p}[f] := \int f(z)\, p(z)\, dz$$

Idea: compute $\int f(z) \frac{p(z)}{q(z)} \cdot q(z) dz =: \mathbb{E}_{z\sim q}\left[f \cdot p/q \right]$

Error depends on $\mathrm{Var}[f \cdot p/q]$, so want $q \approx f \cdot p$ so that it's almost constant (i.e., low variance)

↑ i.e. # of samples

Variants: sampling-importance-resampling (§14.1.6)

- **Markov Chain Monte Carlo** (MCMC) family of methods

History: Metropolis + Ulam 1949

→ Stanislaw Ulam, 1909-1984. Manhattan project
CU math dept. '61-'62, '65-'67, chair '67-'75

Target: $p(z)$

like rejection sampling, we have a proposal distribution q
but now its conditioned on the current state, z^τ,

so our samples $z^{(1)}, z^{(2)}, z^{(3)}, \ldots$ form a Markov chain.

also, as before, assume we can sample from $\tilde{p}$, where $p(z) = \frac{1}{Z_p}\tilde{p}(z)$

- **Metropolis MCMC** (1953) pick a simple $q(\cdot|\cdot)$ and for Metropolis, assume its symmetric: $q(z|\tilde{z}) = q(\tilde{z}|z)$

Given z^τ,

- sample $z^* \sim q(z|z^\tau)$
- ACCEPT w/ probability $A(z^*, z^\tau) := \min(1, \frac{\tilde{p}(z^*)}{\tilde{p}(z^\tau)})$

If ACCEPTED, $z^{\tau+1} = z^*$

else, discard z^* and set $z^{\tau+1} = z^\tau$ (unlike pure rejection sampling which loops)

Does a given transition matrix T have a stationary / invariant / equilibrium distribution?
eigenvalue problem, $Tp = p$
Yes, via Perron-Frobenius thm (1907, 1912)

Under some conditions, $\lim_{\tau\to\infty} z^\tau = p(z)$ in distribution ... but we don't have independent samples! ($M \gg 1$)
(in practice, take every M^{th} sample, pretend its independent)

Sampling (p. 3)

Tuesday, November 12, 2024 5:30 PM

- **Metropolis-Hastings MCMC** ... if q isn't symmetric.

  ACCEPT w.p. $A_k(z^*, z^\tau) := \min\left(1, \frac{\tilde{p}(z^*)}{\tilde{p}(z^\tau)} \cdot \frac{q_k(z^\tau \mid z^*)}{q_k(z^* \mid z^\tau)}\right)$

- **Gibbs Sampling MCMC** 1984, special case of Metropolis-Hastings

  see §14.2.4 or wikipedia

- **Ancestral Sampling MCMC** 1990

  see §14.2.5

- more efficient MCMC, two big directions → Hamiltonian MC (aka Hybrid MC)
  → Langevin Sampling (Metropolis-adjusted Langevin algo)

Background: **Energy-based models**

we're looking for a generative model, sometimes unconditional ($p(x)$), othertimes conditional ($p(x|w)$) ← eg. $w = Ax + v$ inverse problem; w = prompt sentence for DALL-E; w = neural net weight

often write $p(x|w) = \frac{1}{Z(w)} e^{-E(x,w)}$ ← energy function (eg. neural net)

← partition function $= \int e^{-E(x,w)} dw$ } difficult, and for conditional models you'd have to recompute for every w

for MLE, neg. likelihood (of $p(x|w)$), plugging in observations $\{x_1, \dots, x_N\}$, is $\min_w \left(-\sum_{n=1}^{N} E(x_n, w) - N \cdot \ln(Z(w))\right)$

Sometimes we could ignore $-N \cdot \log Z$ term

ex: $p(x|w) \sim N(w, \sigma^2 I)$, $w \in \mathbb{R}^d$ The mean w doesn't affect Z so we can treat it as a constant

Othertimes we can't

ex: $p(x|w) \sim N(0, w^2 I)$, $w \in \mathbb{R}$, pdf is $\frac{1}{w\sqrt{2\pi}} e^{-\frac{1}{2}\frac{x^2}{w^2}}$ ← partition function depends on w

we'll talk later about **Score matching** for **diffusion models**

for now, MCMC perspective for both:

1) training (so find ∇_w (-log-likelihood))
2) inference (once w trained, generate sample $x \sim p(x|w)$)

Sampling (p. 4)

Tuesday, November 12, 2024 8:02 PM

$\nabla_w \ln p(x|w) = -\nabla_w E(x,w) - \nabla_w \ln Z(w)$ for a single point

or for a dataset $D = \{x_1, \dots, x_N\}$

$$\nabla_w \mathbb{E}_{x\sim p_D} \ln p(x|w) = -\mathbb{E}_{x\sim p_D} \nabla_w E(x,w) - \nabla_w \ln Z(w)$$

(straightforward (backprop))

fancy way of saying $\mathbb{E}_{x\sim p_D} f(x) := \frac{1}{N}\sum_{n=1}^{N} f(x_n)$; "empirical" law/distribution

$$-\nabla_w \ln Z(w) = \frac{-1}{Z(w)} \cdot \nabla_w Z(w) \quad \text{chain rule}$$

$$= \frac{-1}{Z(w)} \cdot \nabla_w \cdot \int e^{-E(x,w)} dx \quad \text{def'n of } Z$$

$$= \frac{-1}{Z(w)} \int \nabla_w e^{-E(x,w)} dx \quad \text{wishful thinking (or rigorous via DCT if } e^{-E} \text{ converges... which it should)}$$

$$= \frac{-1}{Z(w)} \int e^{-E(x,w)} \cdot \nabla_w (-E(x,w)) dx \quad \text{chain rule again}$$

$$= \int p(x|w) \cdot \nabla_w E(x,w) dx \quad p(x|w) := \frac{1}{Z(w)} e^{-E(x,w)}$$

$$= \mathbb{E}_{x\sim p(x|w)} \nabla_w E(x,w)$$

So ...

$$\nabla_w \mathbb{E}_{x\sim p_D} \ln p(x|w) = -\mathbb{E}_{x\sim p_D} \nabla_w E(x,w) + \mathbb{E}_{x\sim p(x|w)} \nabla_w E(x,w) \quad (*)$$

(backprop; easy: discrete)

(how to sample? Langevin Sampling)

uses Score function

$$S(x,w) := \nabla_x \ln p(x|w) \quad \text{(not } \nabla_w\text{)}$$

$$= -\nabla_x E(x,w) \quad \leftarrow Z \text{ goes away since } \nabla_x Z(w) = 0$$

Algo: Langevin Dynamics

$$x^{t+1} \leftarrow x^t + \eta \underbrace{\nabla_x p(x^t, w)}_{= S(x^t, w)} + \sqrt{2\eta} \cdot \varepsilon^t \qquad \varepsilon^t \sim N(0, I)$$

as $\eta \to 0$, and $t \to \infty$, $x^t \sim p(x)$ } in our case, it's really $x^t \sim p(x|w)$

So, do this to draw several (M) samples $x_m \sim p(x|w)$

Then approximate (*) via $\mathbb{E}_{x\sim p(x|w)} \nabla_w E(x|w) \approx \frac{1}{M}\sum_{m=1}^{M} \nabla_w E(x_m, w)$