Deep Generative Models

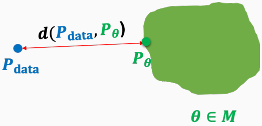
Lecture 13: Energy-Based Models

Aditya Grover

UCLA

Recap of last lecture

$x^{(j)} \sim P_{\text{data}}$
$j = 1, 2, \ldots, m$



- Energy-based models: $p_\theta(\mathbf{x}) = \frac{\exp\{f_\theta(\mathbf{x})\}}{Z(\theta)}$.
  - $Z(\theta)$ is intractable, so no access to likelihood.
  - Comparing the probability of two points is easy: $p_\theta(\mathbf{x}')/p_\theta(\mathbf{x}) = \exp(f_\theta(\mathbf{x}') - f_\theta(\mathbf{x}))$.
- Maximum likelihood training: $\max_\theta\{f_\theta(\mathbf{x}_{train}) - \log Z(\theta)\}$.
  - Contrastive divergence:

$$\nabla_\theta f_\theta(\mathbf{x}_{train}) - \nabla_\theta \log Z(\theta) \approx \nabla_\theta f_\theta(\mathbf{x}_{train}) - \nabla_\theta f_\theta(\mathbf{x}_{sample}),$$

where $\mathbf{x}_{sample} \sim p_\theta(\mathbf{x})$.

Sampling from EBMs: MH-MCMC

Metropolis-Hastings Markov chain Monte Carlo (MCMC).

1. $\mathbf{x}^0 \sim \pi(\mathbf{x})$
2. Repeat for $t = 0, 1, 2, \cdots, T-1$:
   - $\mathbf{x}' = \mathbf{x}^t + \text{noise}$
   - $\mathbf{x}^{t+1} = \mathbf{x}'$ if $f_\theta(\mathbf{x}') \geq f_\theta(\mathbf{x}^t)$
   - If $f_\theta(\mathbf{x}') < f_\theta(\mathbf{x}^t)$, set $\mathbf{x}^{t+1} = \mathbf{x}'$ with probability $\exp\{f_\theta(\mathbf{x}') - f_\theta(\mathbf{x}^t)\}$, otherwise set $\mathbf{x}^{t+1} = \mathbf{x}^t$.

Properties:

- In theory, $\mathbf{x}^T$ converges to $p_\theta(\mathbf{x})$ when $T \to \infty$.
- In practice, need a large number of iterations and convergence slows down exponentially in dimensionality.

Sampling from EBMs: unadjusted Langevin MCMC

Unadjusted Langevin MCMC:

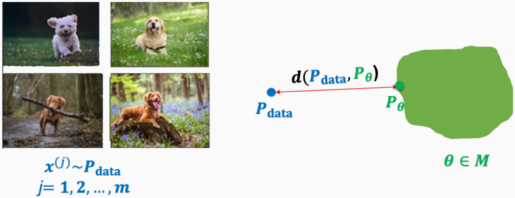
1. $\mathbf{x}^0 \sim \pi(\mathbf{x})$
2. Repeat for $t = 0, 1, 2, \cdots, T-1$:
   - $\mathbf{z}^t \sim \mathcal{N}(0, I)$
   - $\mathbf{x}^{t+1} = \mathbf{x}^t + \epsilon \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^t} + \sqrt{2\epsilon}\mathbf{z}^t$

Properties:

- $\mathbf{x}^T$ converges to $p_\theta(\mathbf{x})$ when $T \to \infty$ and $\epsilon \to 0$.
- $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} f_\theta(\mathbf{x})$ for continuous energy-based models.
- Convergence slows down as dimensionality grows.

Sampling converges slowly in high dimensional spaces and is thus very expensive, yet we need sampling for **each training iteration** in contrastive divergence.

Today's lecture



Goal: Training without sampling

- Score Matching
- Noise Contrastive Estimation
- Adversarial training

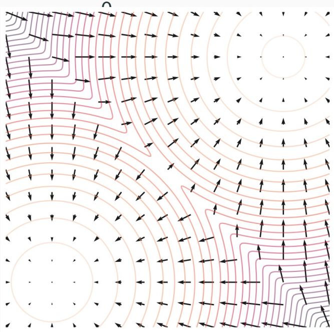
Score function

Energy-based model: $p_\theta(\mathbf{x}) = \frac{\exp\{f_\theta(\mathbf{x})\}}{Z(\theta)}$

(Stein) Score function:

$$s_\theta(\mathbf{x}) := \nabla_\mathbf{x} \log p_\theta(\mathbf{x}) = \nabla_\mathbf{x} f_\theta(\mathbf{x}) - \underbrace{\nabla_\mathbf{x} \log Z(\theta)}_{0} = \nabla_\mathbf{x} f_\theta(\mathbf{x})$$

- Gaussian distribution
  $p_\theta(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
  $\longrightarrow s_\theta(x) = -\frac{x-\mu}{\sigma^2}$
- Gamma distribution
  $p_\theta(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$
  $\longrightarrow s_\theta(x) = \frac{\alpha-1}{x} - \beta$



$p_\theta(\mathbf{x})$ vs. $s_\theta(\mathbf{x})$

Score matching

Observation

$s_\theta(\mathbf{x}) = \nabla_\mathbf{x} \log p_\theta(\mathbf{x})$ is independent of the partition function $Z(\theta)$.

Fisher divergence between $p(\mathbf{x})$ and $q(\mathbf{x})$:

$$D_F(p, q) := \frac{1}{2} E_{\mathbf{x}\sim p}[\|\nabla_\mathbf{x} \log p(\mathbf{x}) - \nabla_\mathbf{x} \log q(\mathbf{x})\|_2^2]$$

Score matching: minimizing the Fisher divergence between $p_{\text{data}}(\mathbf{x})$ and the EBM $p_\theta(\mathbf{x})$

$$\begin{aligned} &\frac{1}{2} E_{\mathbf{x}\sim p_{\text{data}}}[\|\nabla_\mathbf{x} \log p_{\text{data}}(\mathbf{x}) - s_\theta(\mathbf{x})\|_2^2] \\ =& \frac{1}{2} E_{\mathbf{x}\sim p_{\text{data}}}[\|\nabla_\mathbf{x} \log p_{\text{data}}(\mathbf{x}) - \nabla_\mathbf{x} f_\theta(\mathbf{x})\|_2^2] \end{aligned}$$

Score matching

$$\frac{1}{2}E_{\mathbf{x}\sim p_{\text{data}}}[\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})\|_2^2]$$

How to deal with $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$? Integration by parts!

$$\frac{1}{2}E_{x\sim p_{\text{data}}}[(\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x))^2] \quad \text{(Univariate case)}$$

$$= \frac{1}{2}\int p_{\text{data}}(x)[(\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x))^2]\mathrm{d}x$$

$$= \frac{1}{2}\int p_{\text{data}}(x)(\nabla_x \log p_{\text{data}}(x))^2\mathrm{d}x + \frac{1}{2}\int p_{\text{data}}(x)(\nabla_x \log p_\theta(x))^2\mathrm{d}x$$

$$-\int p_{\text{data}}(x)\nabla_x \log p_{\text{data}}(x)\nabla_x \log p_\theta(x)\mathrm{d}x$$

For the cross-correlation term:

$$-\int p_{\text{data}}(x)\nabla_x \log p_{\text{data}}(x)\nabla_x \log p_\theta(x)\mathrm{d}x = -\int p_{\text{data}}(x)\frac{1}{p_{\text{data}}(x)}\nabla_x p_{\text{data}}(x)\nabla_x \log p_\theta(x)\mathrm{d}x$$

$$= \underbrace{-p_{\text{data}}(x)\nabla_x \log p_\theta(x)|_{x=-\infty}^{\infty}}_{=0} + \int p_{\text{data}}(x)\nabla_x^2 \log p_\theta(x)\mathrm{d}x$$

$$= \int p_{\text{data}}(x)\nabla_x^2 \log p_\theta(x)\mathrm{d}x$$

Score matching

Univariate score matching

$$\frac{1}{2}E_{x\sim p_{\text{data}}}[(\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x))^2]$$

$$= \frac{1}{2}\int p_{\text{data}}(x)(\nabla_x \log p_{\text{data}}(x))^2 \mathrm{d}x + \frac{1}{2}\int p_{\text{data}}(x)(\nabla_x \log p_\theta(x))^2 \mathrm{d}x$$

$$- \int p_{\text{data}}(x)\nabla_x \log p_{\text{data}}(x)\nabla_x \log p_\theta(x)\mathrm{d}x$$

$$= \underbrace{\frac{1}{2}\int p_{\text{data}}(x)(\nabla_x \log p_{\text{data}}(x))^2 \mathrm{d}x}_{\text{const.}} + \frac{1}{2}\int p_{\text{data}}(x)(\nabla_x \log p_\theta(x))^2 \mathrm{d}x$$

$$+ \int p_{\text{data}}(x)\nabla_x^2 \log p_\theta(x)\mathrm{d}x$$

$$= E_{x\sim p_{\text{data}}}[\frac{1}{2}(\nabla_x \log p_\theta(x))^2 + \nabla_x^2 \log p_\theta(x)] + \text{const.}$$

Multivariate score matching

$$\frac{1}{2}E_{\mathbf{x}\sim p_{\text{data}}}[\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})\|_2^2]$$

$$= E_{\mathbf{x}\sim p_{\text{data}}}\left[\frac{1}{2}\|\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})\|_2^2 + \text{tr}(\underbrace{\nabla_{\mathbf{x}}^2 \log p_\theta(\mathbf{x})}_{\text{Hessian of } \log p_\theta(\mathbf{x})})\right] + \text{const.}$$

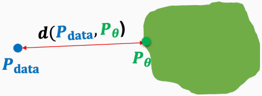
Score matching

1. Sample a mini-batch of datapoints $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \sim p_{\text{data}}(\mathbf{x})$
2. Estimate the score matching loss with the empirical mean

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2}\|\nabla_{\mathbf{x}}\log p_\theta(\mathbf{x}_i)\|_2^2 + \operatorname{tr}(\nabla_{\mathbf{x}}^2 \log p_\theta(\mathbf{x}_i))\right] \\ =&\frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2}\|\nabla_{\mathbf{x}} f_\theta(\mathbf{x}_i)\|_2^2 + \operatorname{tr}(\nabla_{\mathbf{x}}^2 f_\theta(\mathbf{x}_i))\right] \end{aligned}$$

3. Stochastic gradient descent
4. No need to sample from the EBM!

Caveat: Computing the trace of Hessian $\operatorname{tr}(\nabla_{\mathbf{x}}^2 \log p_\theta(\mathbf{x}))$ is in general very expensive for large models. Some solutions: Denoising score matching (Vincent 2010) and sliced score matching (Song et al. 2019).



Distances used for training energy-based models.

- KL divergence = maximum likelihood.

$$\nabla_\theta f_\theta(\mathbf{x}_{data}) - f_\theta(\mathbf{x}_{sample}) \quad \text{(contrastive divergence)}$$

- Fisher divergence = score matching.

$$\frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}}[\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} f_\theta(\mathbf{x})\|_2^2]$$

Noise contrastive estimation

Learning an energy-based model by contrasting it with a noise distribution.

- Data distribution: $p_{\text{data}}(\mathbf{x})$.
- Noise distribution: $p_n(\mathbf{x})$. Should be analytically tractable and easy to sample from.
- Training a discriminator $D_\theta(\mathbf{x}) \in [0, 1]$ to distinguish between data samples and noise samples.

$$\max_\theta E_{\mathbf{x}\sim p_{\text{data}}}[\log D_\theta(\mathbf{x})] + E_{\mathbf{x}\sim p_n}[\log(1 - D_\theta(\mathbf{x}))]$$

- Optimal discriminator $D_{\theta^*}(\mathbf{x})$.

$$D_{\theta^*}(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_n(\mathbf{x})}$$

Noise contrastive estimation

What if the discriminator is parameterized by

$$D_\theta(\mathbf{x}) = \frac{p_\theta(\mathbf{x})}{p_\theta(\mathbf{x}) + p_n(\mathbf{x})}$$

The optimal discriminator $D_{\theta^*}(\mathbf{x})$ satisfies

$$D_{\theta^*}(\mathbf{x}) = \frac{p_{\theta^*}(\mathbf{x})}{p_{\theta^*}(\mathbf{x}) + p_n(\mathbf{x})} = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_n(\mathbf{x})}$$

Equivalently,

$$p_{\theta^*}(\mathbf{x}) = \frac{p_n(\mathbf{x}) D_{\theta^*}(\mathbf{x})}{1 - D_{\theta^*}(\mathbf{x})} = p_{\text{data}}(\mathbf{x})$$

Noise contrastive estimation for training EBMs

Energy-based model:

$$p_\theta(\mathbf{x}) = \frac{e^{f_\theta(\mathbf{x})}}{Z(\theta)}$$

The constraint $Z(\theta) = \int e^{f_\theta(\mathbf{x})}\mathrm{d}\mathbf{x}$ is hard to satisfy.

Solution: Modeling $Z(\theta)$ with an additional trainable parameter Z that *disregards* the constraint $Z = \int e^{f_\theta(\mathbf{x})}\mathrm{d}\mathbf{x}$.

$$p_{\theta,Z}(\mathbf{x}) = \frac{e^{f_\theta(\mathbf{x})}}{Z}$$

The optimal parameters θ^*, Z^* in noise contrastive estimation are

$$p_{\theta^*,Z^*}(\mathbf{x}) = \frac{e^{f_{\theta^*}(\mathbf{x})}}{Z^*} = p_{\text{data}}(\mathbf{x})$$

The optimal parameter Z^* is the correct partition function, because

$$\int \frac{e^{f_{\theta^*}(\mathbf{x})}}{Z^*}\mathrm{d}\mathbf{x} = \int p_{\text{data}}(\mathbf{x})\mathrm{d}\mathbf{x} = 1 \implies Z^* = \int e^{f_{\theta^*}(\mathbf{x})}\mathrm{d}\mathbf{x}$$

Noise contrastive estimation for training EBMs

The discriminator $D_{\theta,Z}(\mathbf{x})$ for probabilistic model $p_{\theta,Z}(\mathbf{x})$ is

$$D_{\theta,Z}(\mathbf{x}) = \frac{\frac{e^{f_\theta(\mathbf{x})}}{Z}}{\frac{e^{f_\theta(\mathbf{x})}}{Z} + p_n(\mathbf{x})} = \frac{e^{f_\theta(\mathbf{x})}}{e^{f_\theta(\mathbf{x})} + p_n(\mathbf{x})Z}$$

Noise contrastive estimation training

$$\max_{\theta,Z} E_{\mathbf{x}\sim p_{\text{data}}}[\log D_{\theta,Z}(\mathbf{x})] + E_{\mathbf{x}\sim p_n}[\log(1 - D_{\theta,Z}(\mathbf{x}))]$$

Equivalently,

$$\begin{aligned}\max_{\theta,Z} E_{\mathbf{x}\sim p_{\text{data}}}[f_\theta(\mathbf{x}) - \log(e^{f_\theta(\mathbf{x})} + Zp_n(\mathbf{x}))] \\ + E_{\mathbf{x}\sim p_n}[\log(Zp_n(\mathbf{x})) - \log(e^{f_\theta(\mathbf{x})} + Zp_n(\mathbf{x}))]\end{aligned}$$

Log-sum-exp trick for numerical stability:

$$\begin{aligned}\log(e^{f_\theta(\mathbf{x})} + Zp_n(\mathbf{x})) &= \log(e^{f_\theta(\mathbf{x})} + e^{\log Z + \log p_n(\mathbf{x})}) \\ &= \text{logsumexp}(f_\theta(\mathbf{x}), \log Z + \log p_n(\mathbf{x}))\end{aligned}$$

Noise contrastive estimation for training EBMs

1. Sample a mini-batch of datapoints $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n \sim p_{\text{data}}(\mathbf{x})$.
2. Sample a mini-batch of noise samples $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n \sim p_n(\mathbf{y})$.
3. Estimate the NCE loss

$$\frac{1}{n}\sum_{i=1}^{n}[f_\theta(\mathbf{x}_i) - \text{logsumexp}(f_\theta(\mathbf{x}_i), \log Z + \log p_n(\mathbf{x}_i))$$

$$+ \log Z + p_n(\mathbf{y}_i) - \text{logsumexp}(f_\theta(\mathbf{y}_i), \log Z + \log p_n(\mathbf{y}_i))]$$

4. Stochastic gradient ascent
5. No need to sample from the EBM!

Comparing NCE and GAN

Similarities:

- Both involve training a discriminator to perform binary classification with a cross-entropy loss
- Both are likelihood-free

Differences:

- Unlike NCE, GAN requires adversarial training or minimax optimization for training
- NCE requires the likelihood of the noise distribution for training, while GAN only requires efficient sampling from the prior
- NCE trains an energy-based model, while GAN trains a deterministic sample generator

Flow contrastive estimation (Gao et al. 2020)

Observations:

- We need to both evaluate the probability of $p_n(\mathbf{x})$, and sample from it efficiently
- We hope to make the classification task as hard as possible, i.e., $p_n(\mathbf{x})$ should be close to $p_{\text{data}}(\mathbf{x})$ (but not exactly the same)

Flow contrastive estimation:

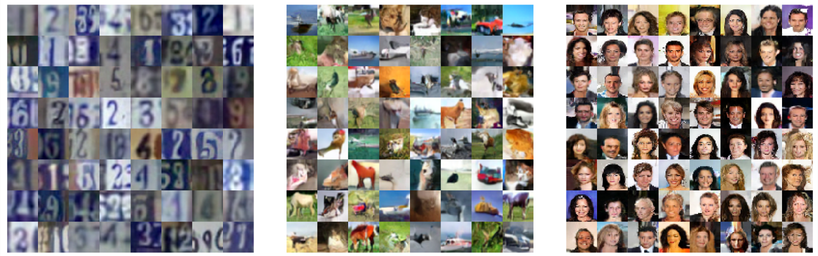
- Parameterize the noise as a normalizing flow $p_{n,\phi}(\mathbf{x})$
- Parameterize the discriminator $D_{\theta,Z,\phi}(\mathbf{x})$ as

$$D_{\theta,Z,\phi}(\mathbf{x}) = \frac{\frac{e^{f_\theta(\mathbf{x})}}{Z}}{\frac{e^{f_\theta(\mathbf{x})}}{Z} + p_{n,\phi}(\mathbf{x})} = \frac{e^{f_\theta(\mathbf{x})}}{e^{f_\theta(\mathbf{x})} + p_{n,\phi}(\mathbf{x})Z}$$

- Train the flow model to minimize $D_{JS}(p_{\text{data}}, p_{n,\phi})$:

$$\min_{\phi} \max_{\theta,Z} E_{\mathbf{x}\sim p_{\text{data}}}[\log D_{\theta,Z,\phi}(\mathbf{x})] + E_{\mathbf{x}\sim p_{n,\phi}}[\log(1 - D_{\theta,Z,\phi}(\mathbf{x}))]$$

Flow contrastive estimation (Gao et al. 2020)



Samples from SVHN, CIFAR-10, and CelebA datasets

Image source: Gao et al. 2020

Adversarial training for EBMs

Energy-based model:

$$p_\theta(\mathbf{x}) = \frac{e^{f_\theta(\mathbf{x})}}{Z(\theta)}$$

Upper bound the log-likelihood with a variational distribution $q_\phi(\mathbf{x})$:

$$\begin{aligned}
E_{\mathbf{x}\sim p_{\text{data}}}[\log p_\theta(\mathbf{x})] &= E_{\mathbf{x}\sim p_{\text{data}}}[f_\theta(\mathbf{x})] - \log Z(\theta) \\
&= E_{\mathbf{x}\sim p_{\text{data}}}[f_\theta(\mathbf{x})] - \log \int e^{f_\theta(\mathbf{x})} \mathrm{d}\mathbf{x} \\
&= E_{\mathbf{x}\sim p_{\text{data}}}[f_\theta(\mathbf{x})] - \log \int q_\phi(\mathbf{x}) \frac{e^{f_\theta(\mathbf{x})}}{q_\phi(\mathbf{x})} \mathrm{d}\mathbf{x} \\
&\leq E_{\mathbf{x}\sim p_{\text{data}}}[f_\theta(\mathbf{x})] - \int q_\phi(\mathbf{x}) \log \frac{e^{f_\theta(\mathbf{x})}}{q_\phi(\mathbf{x})} \mathrm{d}\mathbf{x} \\
&= E_{\mathbf{x}\sim p_{\text{data}}}[f_\theta(\mathbf{x})] - E_{\mathbf{x}\sim q_\phi}[f_\theta(\mathbf{x})] + H(q_\phi(\mathbf{x}))
\end{aligned}$$

Adversarial training

$$\max_\theta \min_\phi E_{\mathbf{x}\sim p_{\text{data}}}[f_\theta(\mathbf{x})] - E_{\mathbf{x}\sim q_\phi}[f_\theta(\mathbf{x})] + H(q_\phi(\mathbf{x}))$$

What do we require for the model $q_\phi(\mathbf{x})$?

Conclusion

- Energy-based models are very flexible probabilistic models with intractable partition functions
- Computing the likelihood is hard
- Comparing the likelihood/probability of two different points is tractable
- Sampling is hard and typically requires iterative MCMC approaches
- Maximum likelihood training by contrastive divergence. Requires sampling for each training iteration
- Sampling-free training methods: score matching, noise contrastive estimation (with partition function estimation), adversarial optimization.
- Reference: *How to Train Your Energy-Based Models* by Yang Song and Durk Kingma