Energy-Based Models

Stefano Ermon, Yang Song

Stanford University

Lecture 12
Energy-based models: \( p_\theta(x) = \frac{\exp\{f_\theta(x)\}}{Z(\theta)} \).

- \( Z(\theta) \) is intractable, so no access to likelihood.
- Comparing the probability of two points is easy:
  \( p_\theta(x')/p_\theta(x) = \exp(f_\theta(x') - f_\theta(x)) \).

Maximum likelihood training: \( \max_{\theta}\{f_\theta(x_{\text{train}}) - \log Z(\theta)\} \).

Contrastive divergence:

\[ \nabla_\theta f_\theta(x_{\text{train}}) - \nabla_\theta \log Z(\theta) \approx \nabla_\theta f_\theta(x_{\text{train}}) - \nabla_\theta f_\theta(x_{\text{sample}}), \]

where \( x_{\text{sample}} \sim p_\theta(x) \).
Sampling from EBMs: MH-MCMC

Metropolis-Hastings Markov chain Monte Carlo (MCMC).

1. $x^0 \sim \pi(x)$

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

Properties:

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

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

Properties:
- \( x^T \) converges to \( p_\theta(x) \) when \( T \to \infty \) and \( \epsilon \to 0 \).
- \( \nabla_x \log p_\theta(x) = \nabla_x f_\theta(x) \) 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.
Goal: Training without sampling

- Score Matching
- Noise Contrastive Estimation
- Adversarial training
Score function

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

**(Stein) Score function:**

\[
s_\theta(x) := \nabla_x \log p_\theta(x) = \nabla_x f_\theta(x) - \underbrace{\nabla_x \log Z(\theta)}_{=0} = \nabla_x f_\theta(x)
\]

- **Gaussian distribution**
  \[
p_\theta(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
  \]
  \[\rightarrow s_\theta(x) = -\frac{x-\mu}{\sigma^2}\]

- **Gamma distribution**
  \[
p_\theta(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}
  \]
  \[\rightarrow s_\theta(x) = \alpha - 1 - \frac{1}{x} - \beta\]
Score matching

**Observation**

$s_\theta(x) = \nabla_x \log p_\theta(x)$ is independent of the partition function $Z(\theta)$.

**Fisher divergence between $p(x)$ and $q(x)$:**

$$D_F(p, q) := \frac{1}{2} \mathbb{E}_{x \sim p} [\| \nabla_x \log p(x) - \nabla_x \log q(x) \|^2]$$

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

$$\frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}} [\| \nabla_x \log p_{\text{data}}(x) - s_\theta(x) \|^2]$$

$$= \frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}} [\| \nabla_x \log p_{\text{data}}(x) - \nabla_x f_\theta(x) \|^2]$$
Score matching

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

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

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

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

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

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

For the cross-correlation term:

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

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

$$= -p_{\text{data}}(x) \nabla_x \log p_{\theta}(x) \bigg|_{x=-\infty}^{\infty} + \int p_{\text{data}}(x) \nabla_x^2 \log p_{\theta}(x) dx$$

$$= 0 + \int p_{\text{data}}(x) \nabla_x^2 \log p_{\theta}(x) dx$$

$$= \int p_{\text{data}}(x) \nabla_x^2 \log p_{\theta}(x) dx$$
Score matching

Univariate score matching

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

Multivariate score matching

\[
\frac{1}{2} E_{x \sim p_{\text{data}}}[\|\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x)\|^2]
\]
\[
= E_{x \sim p_{\text{data}}}\left[\frac{1}{2} \|\nabla_x \log p_\theta(x)\|^2 + \text{tr}(\underbrace{\nabla_x^2 \log p_\theta(x)}_{\text{Hessian of } \log p_\theta(x)})\right] + \text{const.}
\]
Score matching

1. Sample a mini-batch of datapoints \( \{x_1, x_2, \cdots, x_n\} \sim p_{\text{data}}(x) \).
2. Estimate the score matching loss with the empirical mean

\[
\frac{1}{n} \sum_{i=1}^{n} \left[ \frac{1}{2} \| \nabla_x \log p_\theta(x_i) \|_2^2 + \text{tr}(\nabla_x^2 \log p_\theta(x_i)) \right]
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} \left[ \frac{1}{2} \| \nabla_\theta f_\theta(x_i) \|_2^2 + \text{tr}(\nabla_x^2 f_\theta(x_i)) \right]
\]

4. No need to sample from the EBM!

Caveat

Computing the trace of Hessian \( \text{tr}(\nabla_x^2 \log p_\theta(x)) \) is in general very expensive for large models.

Denoising score matching (Vincent 2010) and sliced score matching (Song et al. 2019).
Score matching for learning implicit VAEs

- **Model:** \( p(z), p_\theta(x \mid z), q_\phi(z \mid x) = \delta(z = f_\phi(x, \epsilon)) \).
- **Goal:** Maximize the evidence lower bound (ELBO):

\[
E_{z \sim q_\phi(z \mid x)} [\log p_\theta(x \mid z) p(z)] - E_{z \sim q_\phi(z \mid x)} \log q_\phi(z \mid x) := H(q_\phi(z \mid x))
\]

- Estimate the gradient of the entropy term by training an energy-based model.

\[
\nabla_\phi H(q_\phi(z \mid x)) \\
= - \nabla_\phi E_{z \sim q_\phi(z \mid x)} [\log q_\phi(z \mid x)] \\
= - \nabla_\phi E_\epsilon [\log q_\phi(f_\phi(x, \epsilon) \mid x)] \\
= - E_\epsilon [\nabla z \log q_\phi(z \mid x) |_{z = f_\phi(x, \epsilon)} \nabla_\phi f_\phi(x, \epsilon)]
\]

Score function of \( q_\phi(z \mid x) \)
Score matching for learning implicit VAEs

Samples on CelebA $64 \times 64$.

Image source: Song et al., 2019.
Recap.

Distances used for training energy-based models.

- **KL divergence** = maximum likelihood.
  \[
  \nabla_\theta f_\theta(x_{data}) - f_\theta(x_{sample}) \quad \text{(contrastive divergence)}
  \]

- **Fisher divergence** = score matching.
  \[
  \frac{1}{2} \mathbb{E}_{x \sim p_{data}}[\|\nabla_x \log p_{data}(x) - \nabla_x f_\theta(x)\|_2^2]
  \]
Noise contrastive estimation

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

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

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

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

$$D_\theta^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_n(x)}$$
What if the discriminator is parameterized by

\[ D_\theta(x) = \frac{p_\theta(x)}{p_\theta(x) + p_n(x)} \]

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

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

Equivalently,

\[ p_{\theta^*}(x) = \frac{p_n(x)D_{\theta^*}(x)}{1 - D_{\theta^*}(x)} = p_{\text{data}}(x) \]
Noise contrastive estimation for training EBMs

Energy-based model:

\[ p_\theta(x) = \frac{e^{f_\theta(x)}}{Z(\theta)} \]

The constraint \( Z(\theta) = \int e^{f_\theta(x)} \, dx \) is hard to satisfy. **Solution:** Modeling \( Z(\theta) \) with an additional trainable parameter \( Z \) that disregards the constraint \( Z = \int e^{f_\theta(x)} \, dx \).

\[ p_{\theta, Z}(x) = \frac{e^{f_\theta(x)}}{Z} \]

With noise contrastive estimation, the optimal parameters \( \theta^*, Z^* \) are

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

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

\[ \int \frac{e^{f_{\theta^*}(x)}}{Z^*} \, dx = \int p_{\text{data}}(x) \, dx = 1 \implies Z^* = \int e^{f_{\theta^*}(x)} \, dx \]
Noise contrastive estimation for training EBMs

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

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

Noise contrastive estimation training

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

Equivalently,

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

$$+ E_{x \sim p_n} [\log(Zp_n(x)) - \log(e^{f_\theta(x)} + Zp_n(x))]$$

Log-sum-exp trick for numerical stability:

$$\log(e^{f_\theta(x)} + Zp_n(x)) = \log(e^{f_\theta(x)} + e^{\log Z + \log p_n(x)})$$

$$= \log\text{sumexp}(f_\theta(x), \log Z + \log p_n(x))$$
Noise contrastive estimation for training EBMs

1. Sample a mini-batch of datapoints $x_1, x_2, \cdots, x_n \sim p_{\text{data}}(x)$.
2. Sample a mini-batch of noise samples $y_1, y_2, \cdots, y_n \sim p_n(y)$.
3. Estimate the NCE loss.

$$\frac{1}{n} \sum_{i=1}^{n} [f_{\theta}(x_i) - \log\text{sumexp}(f_{\theta}(x_i), \log Z + \log p_n(x_i))$$

$$+ \log Z + p_n(y_i) - \log\text{sumexp}(f_{\theta}(y_i), \log Z + \log p_n(y_i))]$$

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:
- GAN requires adversarial training or minimax optimization for training, while NCE does not.
- 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 \( p_n(x) \), and sample from it efficiently.
- We hope to make the classification task as hard as possible, i.e., \( p_n(x) \) should be close to \( p_{\text{data}}(x) \) (but not exactly the same).

Flow contrastive estimation:

- Parameterize the noise distribution with a normalizing flow model \( p_{n,\phi}(x) \).
- Parameterize the discriminator \( D_{\theta,Z,\phi}(x) \) as

\[
D_{\theta,Z,\phi}(x) = \frac{e^{\theta}(x)}{Z} + p_{n,\phi}(x) = \frac{e^{\theta}(x)}{e^{\theta}(x) + p_{n,\phi}(x)Z}
\]

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

\[
\min_{\phi} \max_{\theta,Z} \mathbb{E}_{x \sim p_{\text{data}}} [\log D_{\theta,Z,\phi}(x)] + \mathbb{E}_{x \sim p_{n,\phi}} [\log (1 - D_{\theta,Z,\phi}(x))]
\]
Flow contrastive estimation (Gao et al. 2020)

Samples from SVHN, CIFAR-10, and CelebA datasets.

Adversarial training for EBMs

Energy-based model:

\[ p_\theta(x) = \frac{e^{f_\theta(x)}}{Z(\theta)} \]

Upper bounding its log-likelihood with a variational distribution \( q_\phi(x) \):

\[
E_{x \sim p_{\text{data}}} [\log p_\theta(x)] = E_{x \sim p_{\text{data}}} [f_\theta(x)] - \log Z(\theta) \\
= E_{x \sim p_{\text{data}}} [f_\theta(x)] - \log \int e^{f_\theta(x)} \, dx \\
= E_{x \sim p_{\text{data}}} [f_\theta(x)] - \log \int q_\phi(x) \frac{e^{f_\theta(x)}}{q_\phi(x)} \, dx \\
\leq E_{x \sim p_{\text{data}}} [f_\theta(x)] - \int q_\phi(x) \log \frac{e^{f_\theta(x)}}{q_\phi(x)} \, dx \\
= E_{x \sim p_{\text{data}}} [f_\theta(x)] - \int q_\phi(x) [f_\theta(x)] + H(q_\phi(x))
\]

Adversarial training

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

What do we require for the model \( q_\phi(x) \)?
Conclusion

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