Deep Generative Models

11. Energy-Based Models

- 국가수리과학연구소 산업수학혁신센터 김민중

Recap. of Energy-based model

- Energy-based models: $\frac{1}{Z(\theta)}\exp\left(f_\theta(\boldsymbol{x})\right)$
  - $Z(\theta)$ is intractable, so no access to likelihood
  - Comparing the probability of two points is easy

$$\frac{p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{x}')} = \exp\left(f_\theta(\boldsymbol{x}) - f_\theta(\boldsymbol{x}')\right)$$

- Maximum likelihood training:

$$\max_\theta [f_\theta(\boldsymbol{x}_{train}) - \log Z(\theta)]$$

  - Contrastive divergence:

$$\nabla_\theta f_\theta(\boldsymbol{x}_{train}) - \nabla_\theta \log Z(\theta) \approx \nabla_\theta f_\theta(\boldsymbol{x}_{train}) - \nabla_\theta f_\theta\left(\boldsymbol{x}_{sample}\right)$$

  - where $\boldsymbol{x}_{sample} \sim p_\theta(\boldsymbol{x}) = \frac{\exp\left(f_\theta(\boldsymbol{x})\right)}{Z(\theta)}$

Sampling from EBMs: MH-MCMC

- Metropolis-Hastings Markov Chain Monte Carlo (MCMC):
  - Initialize $\boldsymbol{x^0}$ randomly, $t = 0$
  - $\boldsymbol{x'} = \boldsymbol{x}^t + noise$
  - $\boldsymbol{x}^{t+1} = \boldsymbol{x'}$ if $f_\theta(\boldsymbol{x'}) \geq f_\theta(\boldsymbol{x}^t)$
  - If $f_\theta(\boldsymbol{x'}) < f_\theta(\boldsymbol{x}^t)$, set $\boldsymbol{x}^{t+1} = \boldsymbol{x'}$ with probability $\exp\big(f_\theta(x') - f_\theta(x^t)\big)$, otherwise set $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t$
- In theory, when $T \to \infty$, then $\boldsymbol{x}^T \sim p_\theta$
- In practice, we need many iterations, and it converges slowly

Sampling from EBMs: unadjusted Langevin MCMC

- Unadjusted Langevin MCMC
  - Initialize $\boldsymbol{x}^0 \sim \pi(\boldsymbol{x})$
  - Repeat for $t = 0, \ldots, T-1$
    - $\boldsymbol{z} \sim N(0, I)$
    - $\boldsymbol{x}^{t+1} \sim \boldsymbol{x}^t + \epsilon \nabla_x \log p_\theta(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}^t} + \sqrt{2\epsilon}\boldsymbol{z}$
- Properties:
  - If $\epsilon \to 0$ and $T \to \infty$, then we have $\boldsymbol{x}^T$ converges to a sample from p_θ
  - $\nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f_\theta(\boldsymbol{x})$ for continuous energy-based models
  - Convergence slows down as dimensionality grows

Sampling from EBMs: unadjusted Langevin MCMC

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

Today's topic

- Goal: Training method without sampling
  - Score Matching
  - Noise Contrastive Estimation

Score function

- Energy-based models

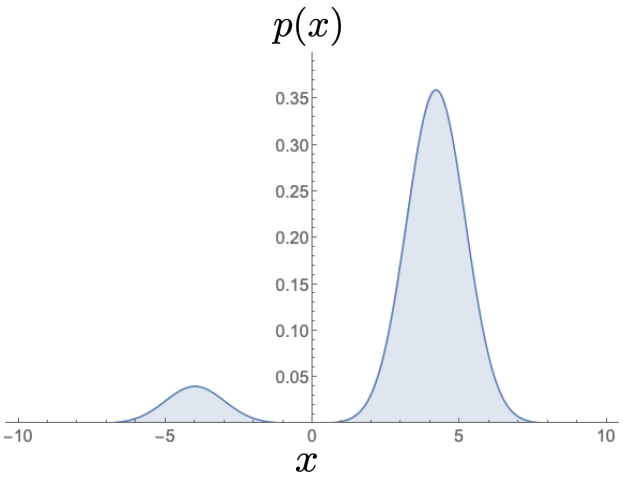
$$\frac{1}{Z(\theta)}\exp\big(f_\theta(\boldsymbol{x})\big), \qquad \log p_\theta(\boldsymbol{x}) = f_\theta(\boldsymbol{x}) - \log Z(\theta)$$

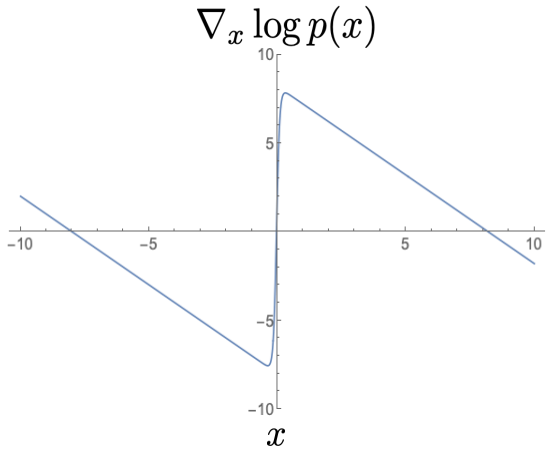
- (Stein) Score function

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

Score matching

- Observation
  - $s_\theta(\boldsymbol{x})$ is independent of the partition function $Z(\theta)$





Score matching

- **Fisher divergence** between $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$:

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

- Score matching
  - minimizing the Fisher divergence between $p_{data}(\boldsymbol{x})$ and the EBM $p_\theta(\boldsymbol{x}) \propto \exp[f_\theta(\boldsymbol{x})]$

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

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

Score matching

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

- How to deal with $\nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x})$ given only samples?
- Integration by parts (univariate case)

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

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

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

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

Score matching

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

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

$$= \underbrace{-p_{data}(x)\nabla_x f_\theta(x)\Big|_{x=-\infty}^{x=\infty}}_{=0} + \int p_{data}(x)\nabla_x^2 f_\theta(x)dx$$

$$= \int p_{data}(x)\nabla_x^2 f_\theta(x)dx$$

- Note that we need to assume p_{data} decays sufficiently rapidly, $p_{data}(x) \to 0$ when $x \to \pm\infty$

Score matching

- Univariate score matching

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

$$= \underbrace{\frac{1}{2} \int p_{data}(x) (\nabla_x \log p_{data}(x))^2 dx}_{const.\ wrt\ \theta} + \frac{1}{2} \int p_{data}(x) \left(\nabla_x f_\theta(x) \right)^2 dx$$

$$+ \int p_{data}(x) \nabla_x^2 f_\theta(x) dx$$

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

Score matching

- Multivariate score matching

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

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

Score matching

- Sample a mini-batch of datapoints $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\} \sim p_{data}(\boldsymbol{x})$
- Estimate the score matching loss with the empirical mean

$$\frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2}\|\nabla_{\boldsymbol{x}}\log p_\theta(\boldsymbol{x}_i)\|_2^2 + tr(\nabla_{\boldsymbol{x}}^2 \log p_\theta(\boldsymbol{x}_i))\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2}\|\nabla_{\boldsymbol{x}} f_\theta(\boldsymbol{x}_i)\|_2^2 + tr\left(\nabla_{\boldsymbol{x}}^2 f_\theta(\boldsymbol{x}_i)\right)\right]$$

- Stochastic gradient descent
- No need to sample from the EBM
- **Remark**: Computing the trace of Hessian $tr\left(\nabla_{\boldsymbol{x}}^2 f_\theta(\boldsymbol{x})\right)$ is in general very expensive for large models
- Denoising score matching (Vincent 2010) and sliced score matching (Song et al. 2019)

Recap

- Distances used for training energy-based models
  - KL divergence = maximum likelihood

$$\nabla_\theta \left[f_\theta(\boldsymbol{x}_{data}) - f_\theta(x_{sample}) \right] \quad \text{(contrastive divergence)}$$

- Fisher divergence = score matching

$$\min_\theta E_{\boldsymbol{x} \sim p_{data}} [\, \| \nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} f_\theta(\boldsymbol{x}) \|_2^2]$$

$$\Leftrightarrow \min_\theta E_{\boldsymbol{x} \sim p_{data}} \left[\frac{1}{2} \| \nabla_{\boldsymbol{x}} f_\theta(\boldsymbol{x}) \|_2^2 + tr\left(\nabla_{\boldsymbol{x}}^2 f_\theta(\boldsymbol{x}) \right) \right]$$

Noise contrastive estimation

- Learning an energy-based model by contrasting it with a noise distribution
  - Data distribution: $p_{data}(\boldsymbol{x})$
  - Noise distribution: $p_n(\boldsymbol{x})$ analytically tractable and easy to sample from
  - Training a discriminator(neural network) $D_\theta(\boldsymbol{x}) \in [0,1]$ to distinguish between data samples and noise samples

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

- What is the optimal discriminator $D_{\theta^*}(\boldsymbol{x})$?

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

Noise contrastive estimation

- Let the discriminator be parametrized by $D_\theta(\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{x})+p_n(\boldsymbol{x})}$
- Then the optimal discriminator $D_{\theta^*}(\boldsymbol{x})$ satisfies

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

- By training the discriminator, we are implicitly learning $p_{\theta^*} \approx p_{data}$. Particularly suitable for cases where $p_\theta(\boldsymbol{x})$ is defined up to a normalization constant (EBMs)
- Equivalently,

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

- Classifier is used to correct density estimates from p_n. It can be used to improve a base generative model (Boosted Generative Models, Grover et al., 2018)

Noise contrastive estimation for training EBMs

- Energy-based models: $\frac{1}{Z(\theta)}\exp\left(f_\theta(\boldsymbol{x})\right)$
- Modeling $Z(\theta)$ with an additional trainable parameter Z that is not explicitly constrained to satisfy $Z = \int e^{f_\theta(\boldsymbol{x})} d\boldsymbol{x}$
- Let

$$p_{\theta,Z}(\boldsymbol{x}) = \frac{\exp\left(f_\theta(\boldsymbol{x})\right)}{Z}$$

- With noise contrastive estimation, the optimal parameters θ^*, Z^* are

$$p_{\theta^*,Z^*}(\boldsymbol{x}) = \frac{\exp\left(f_{\theta^*}(\boldsymbol{x})\right)}{Z^*} = p_{data}(\boldsymbol{x})$$

- The optimal parameter Z^*

$$\int \frac{\exp\left(f_{\theta^*}(\boldsymbol{x})\right)}{Z^*} d\boldsymbol{x} = \int p_{data}(\boldsymbol{x}) d\boldsymbol{x} = 1 \implies Z^* = \int \exp\left(f_{\theta^*}(\boldsymbol{x})\right) d\boldsymbol{x}$$

Noise contrastive estimation for training EBMs

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

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

- Noise contrastive estimation training

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

- Equivalently,

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

- Log-sum-exp trick for numerical stability:

$$\log\left(e^{f_\theta(\boldsymbol{x})} + Zp_n(\boldsymbol{x})\right) = \mathrm{logsumexp}(f_\theta(\boldsymbol{x}), \log Z + \log p_n(\boldsymbol{x}))$$

Noise contrastive estimation for training EBMs

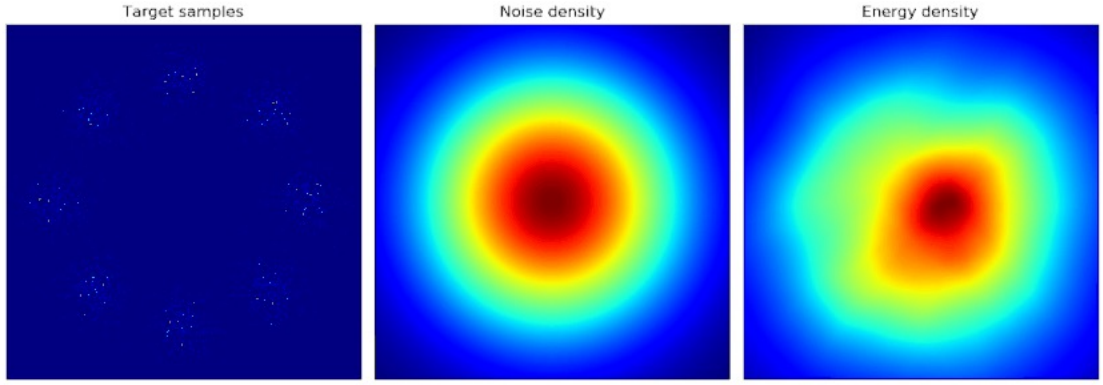
- Sample a mini-batch of datapoints $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(n)} \sim p_{data}(\boldsymbol{x})$
- Sample a mini-batch of datapoints $\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \cdots, \boldsymbol{y}^{(n)} \sim p_n(\boldsymbol{y})$
- Estimate the NCE loss

$$\frac{1}{n}\sum_{i=1}^{n}\left[f_\theta\left(\boldsymbol{x}^{(i)}\right) - \text{logsumexp}\left(f_\theta\left(\boldsymbol{x}^{(i)}\right), \log Z + \log p_n\left(\boldsymbol{x}^{(i)}\right)\right) + \log Z\right.$$
$$\left. + p_n\left(\boldsymbol{y}^{(i)}\right) - \text{logsumexp}\left(f_\theta\left(\boldsymbol{y}^{(i)}\right), \log Z + \log p_n\left(\boldsymbol{y}^{(i)}\right)\right)\right]$$

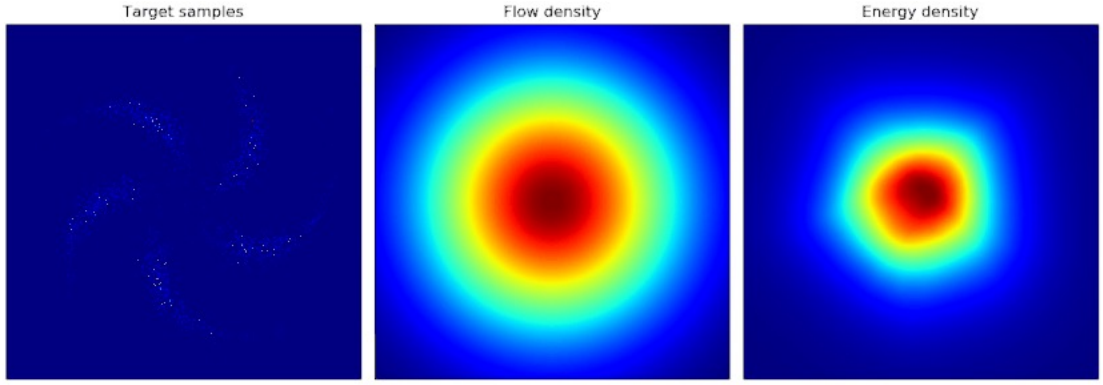
- Stochastic gradient ascent with respect to θ and Z
- No need to sample from the EBM

Noise contrastive estimation

- 8 Gaussians



- Pinwheel



Source: https://github.com/lifeitech/nce

Comparing NCE and GAN

- **Similarities**
  - Both involve training a discriminator to perform binary classification with a cross-entropy loss
  - Both are likelihood-free (recall likelihood not tractable in EBM)
- **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 $p_n(\boldsymbol{x})$, and sample from it efficiently
  - We hope to make the classification task as hard as possible, i.e., $p_n(\boldsymbol{x})$ should be close to $p_{data}(\boldsymbol{x})$ (but not exactly the same)

Flow contrastive estimation (Gao et al. 2020)

- **Flow contrastive estimation**
  - Parameterize the noise distribution with a normalizing flow model $p_{n,\phi}(\boldsymbol{x})$
  - Parameterize the discriminator $D_{\theta,Z,\phi}(\boldsymbol{x})$ as

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

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

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

Flow contrastive estimation (Gao et al. 2020)

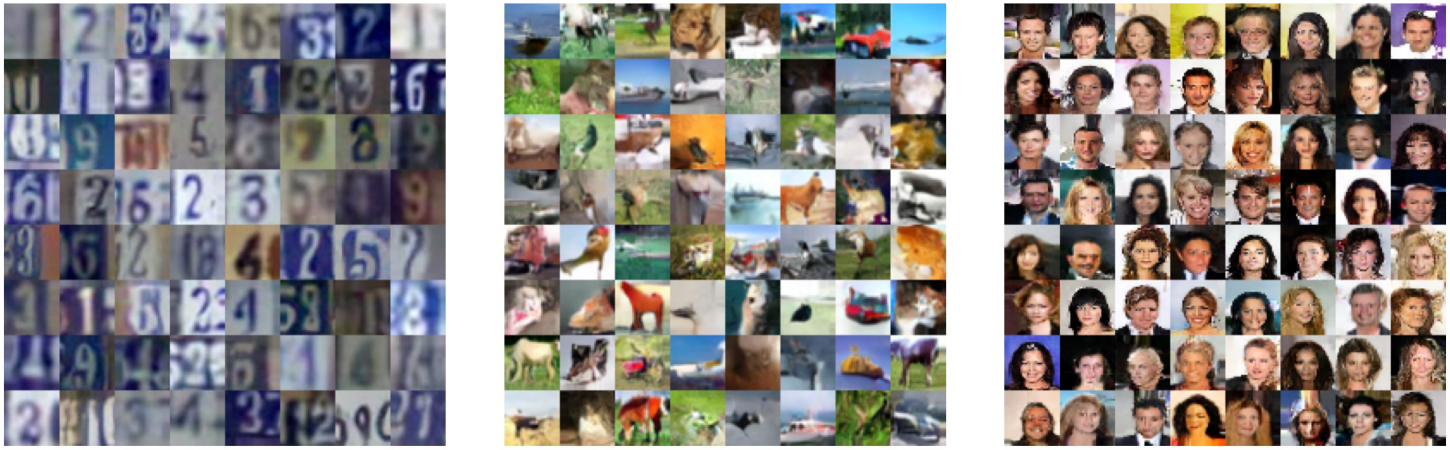


Figure 3: Synthesized examples from the Glow model learned by FCE. From left to right panels are from SVHN, CIFAR-10 and CelebA datasets, respectively. The image size is 32×32.

Source: Gao et al. 2020

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, noise contrastive estimation (can provide an estimate of the partition function)
- Reference: How to Train Your Energy-Based Models by Yang Song and Durk Kingma

Thanks