Probabilistic (Graphical) Models

and inference

Oliver Obst · Autumn 2024

WESTERN SYDNEY
UNIVERSITY

Probabilistic (Graphical) Models and Inference

(PGM: Probabilistic Graphical Models: Principles and Techniques by Daphne Koller and Nir Friedman. MIT Press)
(PMLI: Probabilistic Machine Learning: An introduction by Kevin Murphy. MIT Press)

Week	Lecture	Required reading	Assessment
1 Monday, 4 March 2024	Introduction, Probability Theory	PGM Chapter 2, PMLI Chapter 6.1	
2 Monday, 11 March 2024	Directed and undirected networks introduction	PGM Chapter 3 & 4	Quiz 1
3 Monday, 18 March 2024	Variable elimination	PGM Chapter 9	
4 Monday, 25 March 2024	Belief propagation	PGM Chapter 10/11	Quiz 2
5 Monday, 1 April 2024	public holiday		5 April 2024: census date
6 Monday, 8 April 2024	Message passing / Graph neural networks	https://distill.pub/2021/gnn-intro/	
7 Monday, 15 April 2024	Sampling	PGM Chapter 12	Quiz 3
8 Monday, 22 April 2024	Mid-term break		
9 Monday, 29 April 2024	Variational inference	https://leimao.github.io/article/Introduction-to-Variational-Inference/	Intra-session exam
10 Monday, 6 May 2024	Autoregressive models	https://sites.google.com/view/berkeley-cs294-158-sp20/home	Quiz 4
11 Monday, 13 May 2024	Variational Auto-Encoders	https://lilianweng.github.io/posts/2018-08-12-vae/	
12 Monday, 20 May 2024	GANs	https://arxiv.org/abs/1701.00160	Quiz 5
13 Monday, 27 May 2024	Energy-based models	https://arxiv.org/abs/2101.03288	
14 Monday, 3 June 2024	Evaluating generative models		Quiz 6
Monday, 17 June 2024			Project due

What are energy based models?

What are energy based models?

Suppose we have access to an **unnormalised** distribution $\tilde{p}_\theta$

$$p_\theta(\mathbf{x}) = \frac{\tilde{p}_\theta(\mathbf{x})}{Z_\theta}.$$

We call the negative log of $\tilde{p}_\theta(\mathbf{x})$ its **energy function**

$$E_\theta(\mathbf{x}) = -\log \tilde{p}_\theta(\mathbf{x}).$$

Energy function equal to $-\log p_\theta(\mathbf{x})$ up to a constant independent of $\mathbf{x}$

$$p_\theta(\mathbf{x}) = \frac{e^{-E_\theta(\mathbf{x})}}{Z_\theta} \Rightarrow E_\theta(\mathbf{x}) = -\log p_\theta(\mathbf{x}) - \log Z_\theta,$$

We use the term Energy Based Models (EBM) for energy functions $E_\theta(\mathbf{x})$ where $Z_\theta = \int e^{-E_\theta(\mathbf{x})} d\mathbf{x}$ is not tractable.

Why Energy Based Models?

Energy-based models bring us **flexibility** in

- **Model Design**: EBMs place fewer restrictions on model design compared to other generative models (e.g., Auto-Regressive Models, VAEs, Normalising Flows).

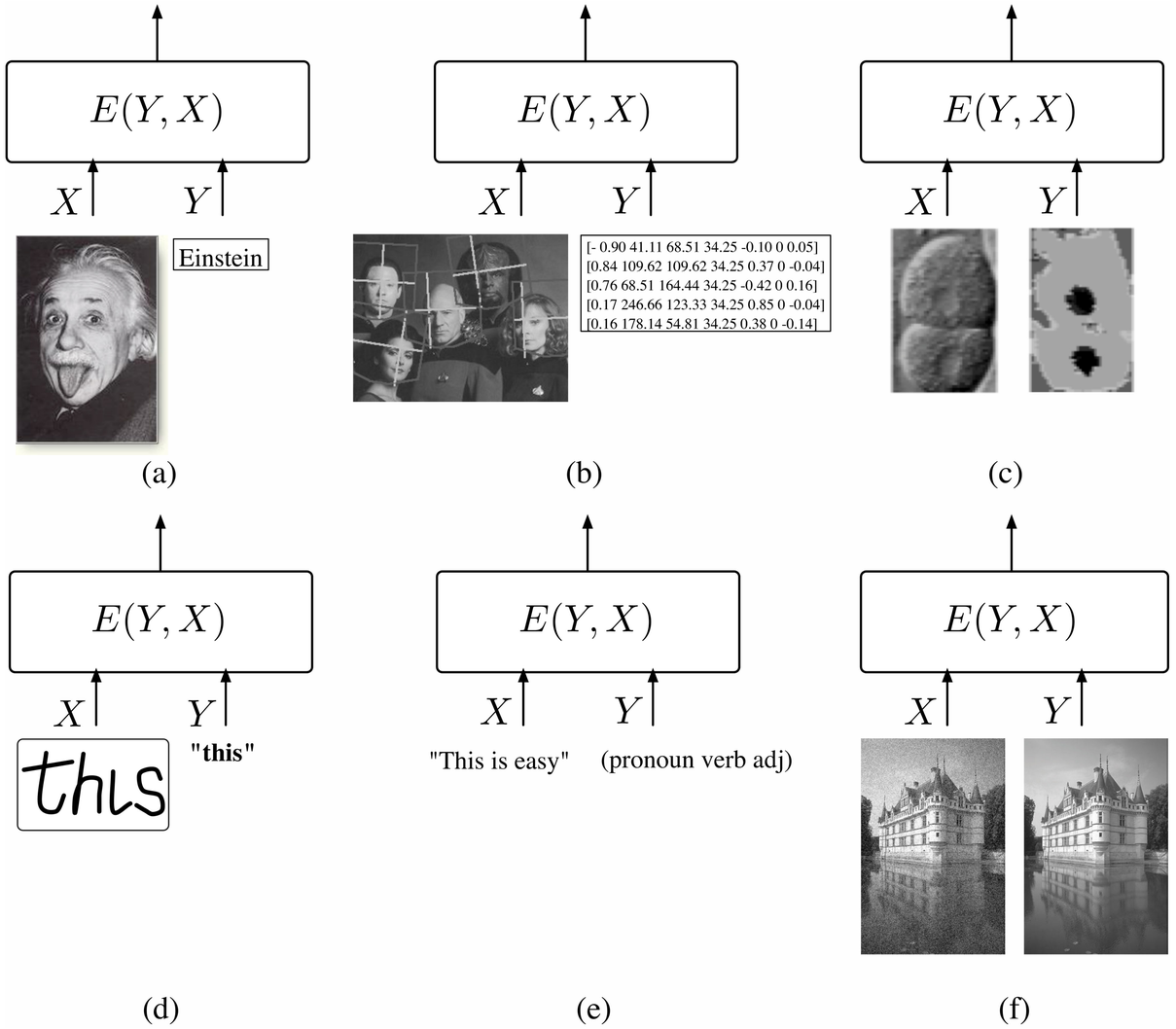
  We need an (almost) arbitrary $E_\theta : \mathcal{X} \to \mathbb{R}$.

  Not constrained to models with tractable likelihoods.

- **Problem Application**: The EBM framework is extremely general.
  If we can rephrase the problem as a scalar function, we can apply EBMs...

Example EBM Applications

(Yann LeCun et al., 2006: A tutorial on energy-based learning)



An example — product of experts [Hinton, 2002]

Product of N experts has the form

An example — product of experts [Hinton, 2002]

Product of N experts has the form

$$p_\theta(\mathbf{x}) = \frac{1}{Z_\theta} \prod_{n=1}^{N} p_{n,\theta}(\mathbf{x}) \Leftrightarrow \log p_\theta(\mathbf{x}) = \underbrace{\sum_{n=1}^{N} \log p_{n,\theta}(\mathbf{x})}_{-E_\theta(\mathbf{x})} - \log Z_\theta$$

An example — product of experts [Hinton, 2002]

Product of N experts has the form

$$p_\theta(\mathbf{x}) = \frac{1}{Z_\theta} \prod_{n=1}^{N} p_{n,\theta}(\mathbf{x}) \Leftrightarrow \log p_\theta(\mathbf{x}) = \underbrace{\sum_{n=1}^{N} \log p_{n,\theta}(\mathbf{x})}_{-E_\theta(\mathbf{x})} - \log Z_\theta$$

We would want to train the model via maximum likelihood

An example — product of experts [Hinton, 2002]

Product of N experts has the form

$$p_\theta(\mathbf{x}) = \frac{1}{Z_\theta} \prod_{n=1}^{N} p_{n,\theta}(\mathbf{x}) \Leftrightarrow \log p_\theta(\mathbf{x}) = \underbrace{\sum_{n=1}^{N} \log p_{n,\theta}(\mathbf{x})}_{-E_\theta(\mathbf{x})} - \log Z_\theta$$

We would want to train the model via maximum likelihood

$$\theta^* = \arg\max_\theta \log p_\theta(\mathbf{x})$$

An example — product of experts [Hinton, 2002]

Product of N experts has the form

$$p_\theta(\mathbf{x}) = \frac{1}{Z_\theta}\prod_{n=1}^{N} p_{n,\theta}(\mathbf{x}) \Leftrightarrow \log p_\theta(\mathbf{x}) = \underbrace{\sum_{n=1}^{N} \log p_{n,\theta}(\mathbf{x})}_{-E_\theta(\mathbf{x})} - \log Z_\theta$$

We would want to train the model via maximum likelihood

$$\theta^* = \arg\max_\theta \log p_\theta(\mathbf{x})$$

But Z_θ is intractable for general $E_\theta(\mathbf{x})$. Take the gradient w.r.t. θ

An example — product of experts [Hinton, 2002]

Product of N experts has the form

$$p_\theta(\mathbf{x}) = \frac{1}{Z_\theta}\prod_{n=1}^{N} p_{n,\theta}(\mathbf{x}) \Leftrightarrow \log p_\theta(\mathbf{x}) = \underbrace{\sum_{n=1}^{N} \log p_{n,\theta}(\mathbf{x})}_{-E_\theta(\mathbf{x})} - \log Z_\theta$$

We would want to train the model via maximum likelihood

$$\theta^* = \arg\max_\theta \log p_\theta(\mathbf{x})$$

But Z_θ is intractable for general $E_\theta(\mathbf{x})$. Take the gradient w.r.t. θ

$$\nabla_\theta \log p_\theta(\mathbf{x}) = -\nabla_\theta E_\theta(\mathbf{x}) - \nabla_\theta \log Z_\theta$$
$$\nabla_\theta \log Z_\theta = -\mathbb{E}_{x\sim p_\theta(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x})]$$

An example — product of experts [Hinton, 2002]

Product of N experts has the form

$$p_\theta(\mathbf{x}) = \frac{1}{Z_\theta}\prod_{n=1}^{N} p_{n,\theta}(\mathbf{x}) \Leftrightarrow \log p_\theta(\mathbf{x}) = \underbrace{\sum_{n=1}^{N} \log p_{n,\theta}(\mathbf{x})}_{-E_\theta(\mathbf{x})} - \log Z_\theta$$

We would want to train the model via maximum likelihood

$$\theta^* = \arg\max_\theta \log p_\theta(\mathbf{x})$$

But Z_θ is intractable for general $E_\theta(\mathbf{x})$. Take the gradient w.r.t. θ

$$\nabla_\theta \log p_\theta(\mathbf{x}) = -\nabla_\theta E_\theta(\mathbf{x}) - \nabla_\theta \log Z_\theta$$
$$\nabla_\theta \log Z_\theta = -\mathbb{E}_{x\sim p_\theta(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x})]$$

Sampling $x \sim p_\theta(\mathbf{x})$ is intractable.

Training PoEs by Contrastive Divergence

Contrastive Divergence is a cancellation trick plus an approximation

Training PoEs by Contrastive Divergence

Contrastive Divergence is a cancellation trick plus an approximation

Let $p^0(\mathbf{x}) = p_D(\mathbf{x})$ be the true data distribution.

$$\theta^* = \arg\max_\theta \log p_\theta(\mathbf{x}) \Leftrightarrow \theta^* = \arg\min_\theta \mathrm{KL}(p^0 \| p_\theta)$$

Let $p_\theta^t(\mathbf{x})$ be the distribution of the data after t steps of MCMC

$$\mathbf{x} \sim p_\theta^t(\mathbf{x}) \Leftrightarrow \mathbf{x} \sim \mathrm{MCMC}(\mathrm{target} = p_\theta, \mathrm{init} = \mathbf{x}_0), \mathbf{x}_0 \sim p^0(\mathbf{x}).$$

Note that $p_\theta^t \to p_\theta$ as $t \to \infty$, so $p_\theta^\infty = p_\theta$.

Idea: Run MCMC for a few iterations ($t = 1$), minimise

$$\Delta\mathrm{KL} = \mathrm{KL}(p^0 \| p_\theta^\infty) - \mathrm{KL}(p_\theta^t \| p_\theta^\infty)$$

Motivation: Pesky term from Z_θ cancels.

Training PoEs by Contrastive Divergence

$$\Delta \mathrm{KL} = \underbrace{\mathrm{KL}(p^0 \| p_\theta^\infty)}_{(1)} - \underbrace{\mathrm{KL}(p_\theta^t \| p_\theta^\infty)}_{(2)}$$

Training PoEs by Contrastive Divergence

$$\Delta \mathrm{KL} = \underbrace{\mathrm{KL}(p^0 \| p_\theta^\infty)}_{(1)} - \underbrace{\mathrm{KL}(p_\theta^t \| p_\theta^\infty)}_{(2)}$$

Property 1: p_θ^t is always closer to p_θ^∞ than p^0 is to p_θ^∞.

Training PoEs by Contrastive Divergence

$$\Delta \mathrm{KL} = \underbrace{\mathrm{KL}(p^0 \| p_\theta^\infty)}_{(1)} - \underbrace{\mathrm{KL}(p_\theta^t \| p_\theta^\infty)}_{(2)}$$

Property 1: p_θ^t is always closer to p_θ^∞ than p^0 is to p_θ^∞.

$$\mathrm{KL}(p_\theta^t \| p_\theta^\infty) \leq \mathrm{KL}(p^0 \| p_\theta^\infty) \Rightarrow \Delta \mathrm{KL} \geq 0$$

Training PoEs by Contrastive Divergence

$$\Delta \mathrm{KL} = \underbrace{\mathrm{KL}(p^0 \| p_\theta^\infty)}_{(1)} - \underbrace{\mathrm{KL}(p_\theta^t \| p_\theta^\infty)}_{(2)}$$

Property 1: p_θ^t is always closer to p_θ^∞ than p^0 is to p_θ^∞.

$$\mathrm{KL}(p_\theta^t \| p_\theta^\infty) \leq \mathrm{KL}(p^0 \| p_\theta^\infty) \Rightarrow \Delta \mathrm{KL} \geq 0$$

Property 2: If p^0 is equal to p_θ^∞, so is p_θ^t

Training PoEs by Contrastive Divergence

$$\Delta \mathrm{KL} = \underbrace{\mathrm{KL}(p^0 \| p_\theta^\infty)}_{(1)} - \underbrace{\mathrm{KL}(p_\theta^t \| p_\theta^\infty)}_{(2)}$$

Property 1: p_θ^t is always closer to p_θ^∞ than p^0 is to p_θ^∞.

$$\mathrm{KL}(p_\theta^t \| p_\theta^\infty) \leq \mathrm{KL}(p^0 \| p_\theta^\infty) \Rightarrow \Delta \mathrm{KL} \geq 0$$

Property 2: If p^0 is equal to p_θ^∞, so is p_θ^t

$$p_\theta^\infty = p^0 \Rightarrow p_\theta^t = p^0, \Delta \mathrm{KL} = 0$$

Training PoEs by Contrastive Divergence

$$\Delta \mathrm{KL} = \underbrace{\mathrm{KL}(p^0 \| p_\theta^\infty)}_{(1)} - \underbrace{\mathrm{KL}(p_\theta^t \| p_\theta^\infty)}_{(2)}$$

Property 1: p_θ^t is always closer to p_θ^∞ than p^0 is to p_θ^∞.

$$\mathrm{KL}(p_\theta^t \| p_\theta^\infty) \leq \mathrm{KL}(p^0 \| p_\theta^\infty) \Rightarrow \Delta \mathrm{KL} \geq 0$$

Property 2: If p^0 is equal to p_θ^∞, so is p_θ^t

$$p_\theta^\infty = p^0 \Rightarrow p_\theta^t = p^0, \Delta \mathrm{KL} = 0$$

Property 3: Running $t \to \infty$ recovers maximum likelihood

Training PoEs by Contrastive Divergence

$$\Delta \mathrm{KL} = \underbrace{\mathrm{KL}(p^0 \| p_\theta^\infty)}_{(1)} - \underbrace{\mathrm{KL}(p_\theta^t \| p_\theta^\infty)}_{(2)}$$

Property 1: p_θ^t is always closer to p_θ^∞ than p^0 is to p_θ^∞.

$$\mathrm{KL}(p_\theta^t \| p_\theta^\infty) \leq \mathrm{KL}(p^0 \| p_\theta^\infty) \Rightarrow \Delta \mathrm{KL} \geq 0$$

Property 2: If p^0 is equal to p_θ^∞, so is p_θ^t

$$p_\theta^\infty = p^0 \Rightarrow p_\theta^t = p^0, \Delta \mathrm{KL} = 0$$

Property 3: Running $t \to \infty$ recovers maximum likelihood

$$\Delta \mathrm{KL} \to \mathrm{KL}(p^0 \| p_\theta^\infty), \text{as } t \to \infty.$$

Training PoEs by Contrastive Divergence

$$\Delta \mathrm{KL} = \underbrace{\mathrm{KL}(p^0 \| p_\theta^\infty)}_{(1)} - \underbrace{\mathrm{KL}(p_\theta^t \| p_\theta^\infty)}_{(2)}$$

Property 1: p_θ^t is always closer to p_θ^∞ than p^0 is to p_θ^∞.

$$\mathrm{KL}(p_\theta^t \| p_\theta^\infty) \leq \mathrm{KL}(p^0 \| p_\theta^\infty) \Rightarrow \Delta \mathrm{KL} \geq 0$$

Property 2: If p^0 is equal to p_θ^∞, so is p_θ^t

$$p_\theta^\infty = p^0 \Rightarrow p_\theta^t = p^0, \Delta \mathrm{KL} = 0$$

Property 3: Running $t \to \infty$ recovers maximum likelihood

$$\Delta \mathrm{KL} \to \mathrm{KL}(p^0 \| p_\theta^\infty), \text{as } t \to \infty.$$

Intuition: ΔKL encourages (1): p_θ^∞ close to p^0 and (2): p_θ^t far from p^0.

Training PoEs by Contrastive Divergence

$$\Delta \mathrm{KL} = \underbrace{\mathrm{KL}(p^0 \| p_\theta^\infty)}_{(1)} - \underbrace{\mathrm{KL}(p_\theta^t \| p_\theta^\infty)}_{(2)}$$

Property 1: p_θ^t is always closer to p_θ^∞ than p^0 is to p_θ^∞.

$$\mathrm{KL}(p_\theta^t \| p_\theta^\infty) \leq \mathrm{KL}(p^0 \| p_\theta^\infty) \Rightarrow \Delta \mathrm{KL} \geq 0$$

Property 2: If p^0 is equal to p_θ^∞, so is p_θ^t

$$p_\theta^\infty = p^0 \Rightarrow p_\theta^t = p^0, \Delta \mathrm{KL} = 0$$

Property 3: Running $t \to \infty$ recovers maximum likelihood

$$\Delta \mathrm{KL} \to \mathrm{KL}(p^0 \| p_\theta^\infty), \text{as } t \to \infty.$$

Intuition: ΔKL encourages (1): p_θ^∞ close to p^0 and (2): p_θ^t far from p^0.

Since p_θ^t starts at p^0, (2) encourages the chain to not wander from p^0.

Training PoEs by Contrastive Divergence

The cancellation leaves us with

Training PoEs by Contrastive Divergence

The cancellation leaves us with

$$\nabla_\theta \Delta \mathrm{KL} = \int \left[p^0 \frac{dE_\theta}{d\theta} - p_\theta^t \frac{dE_\theta}{d\theta} - \frac{dp_\theta^t}{d\theta} \frac{\delta}{\delta p_\theta^t} \mathrm{KL}(p_\theta^t \| p_\theta^\infty) \right] d\mathbf{x}$$

Training PoEs by Contrastive Divergence

The cancellation leaves us with

$$\nabla_\theta \Delta \mathrm{KL} = \int \left[p^0 \frac{dE_\theta}{d\theta} - p_\theta^t \frac{dE_\theta}{d\theta} - \frac{dp_\theta^t}{d\theta} \frac{\delta}{\delta p_\theta^t} \mathrm{KL}(p_\theta^t \| p_\theta^\infty) \right] d\mathbf{x}$$

The first term can be estimated by setting $\mathbf{x}$ equal to the data.

Training PoEs by Contrastive Divergence

The cancellation leaves us with

$$\nabla_\theta \Delta \mathrm{KL} = \int \left[p^0 \frac{dE_\theta}{d\theta} - p_\theta^t \frac{dE_\theta}{d\theta} - \frac{dp_\theta^t}{d\theta} \frac{\delta}{\delta p_\theta^t} \mathrm{KL}(p_\theta^t \| p_\theta^\infty) \right] d\mathbf{x}$$

The first term can be estimated by setting $\mathbf{x}$ equal to the data.

The second term can be estimated with simple Monte Carlo.

Training PoEs by Contrastive Divergence

The cancellation leaves us with

$$\nabla_\theta \Delta \mathrm{KL} = \int \left[p^0 \frac{dE_\theta}{d\theta} - p_\theta^t \frac{dE_\theta}{d\theta} - \frac{dp_\theta^t}{d\theta} \frac{\delta}{\delta p_\theta^t} \mathrm{KL}(p_\theta^t \| p_\theta^\infty) \right] d\mathbf{x}$$

The first term can be estimated by setting $\mathbf{x}$ equal to the data.

The second term can be estimated with simple Monte Carlo.

The last term is still tricky. Hinton [2002] ignores it.

Training PoEs by Contrastive Divergence

The cancellation leaves us with

$$\nabla_\theta \Delta \mathrm{KL} = \int \left[p^0 \frac{dE_\theta}{d\theta} - p_\theta^t \frac{dE_\theta}{d\theta} - \frac{dp_\theta^t}{d\theta} \frac{\delta}{\delta p_\theta^t} \mathrm{KL}(p_\theta^t \| p_\theta^\infty) \right] d\mathbf{x}$$

The first term can be estimated by setting $\mathbf{x}$ equal to the data.

The second term can be estimated with simple Monte Carlo.

The last term is still tricky. Hinton [2002] ignores it.

$$\nabla_\theta \Delta \mathrm{KL} \approx \mathbb{E}_{\mathbf{x} \sim p^0(\mathbf{x})}[\frac{dE_\theta(\mathbf{x})}{d\theta}] - \mathbb{E}_{\mathbf{x} \sim p^t(\mathbf{x})}[\frac{dE_\theta(\mathbf{x})}{d\theta}].$$

Training PoEs by Contrastive Divergence

The cancellation leaves us with

$$\nabla_\theta \Delta \mathrm{KL} = \int \left[p^0 \frac{dE_\theta}{d\theta} - p^t_\theta \frac{dE_\theta}{d\theta} - \frac{dp^t_\theta}{d\theta} \frac{\delta}{\delta p^t_\theta} \mathrm{KL}(p^t_\theta \| p^\infty_\theta) \right] d\mathbf{x}$$

The first term can be estimated by setting $\mathbf{x}$ equal to the data.

The second term can be estimated with simple Monte Carlo.

The last term is still tricky. Hinton [2002] ignores it.

$$\nabla_\theta \Delta \mathrm{KL} \approx \mathbb{E}_{\mathbf{x} \sim p^0(\mathbf{x})}[\frac{dE_\theta(\mathbf{x})}{d\theta}] - \mathbb{E}_{\mathbf{x} \sim p^t(\mathbf{x})}[\frac{dE_\theta(\mathbf{x})}{d\theta}].$$

Empirically shows that this update also reduces the ignored term.

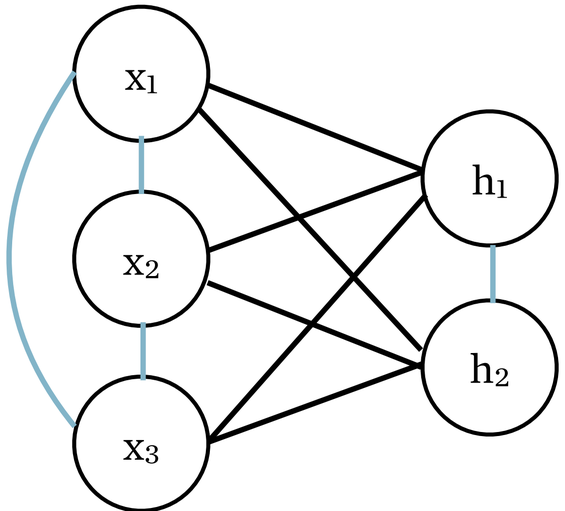
(Restricted) Boltzmann Machines

Boltzmann Machines [BM; Hinton et al., 1986], early example of EBM

$$E_\theta(\mathbf{x}, \mathbf{h}) = \sum_{i \in X, j \in H} x_i h_j w_{ij} + \sum_{i \in X} x_i \theta_i + \sum_{i \in H} h_i \theta_i$$

$$+ \sum_{i<j \in X} x_i x_j w_{ij} + \sum_{i<j \in H} h_i h_j w_{ij} \,.$$

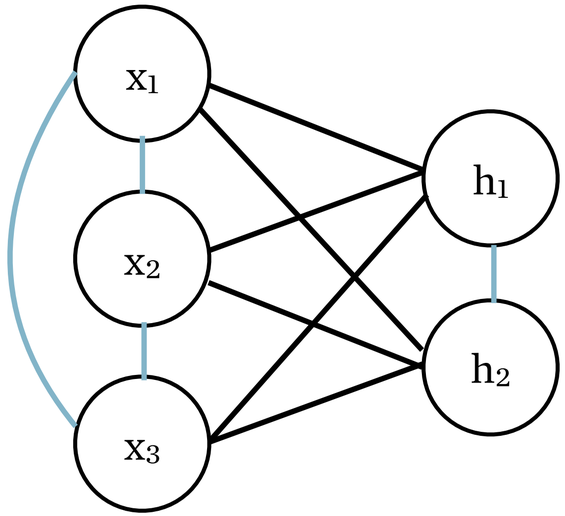
where $x_i, h_i \in \{0,1\}$. Smolensky [1986] introduced restricted BMs.



Graphical model of a (restricted) Boltzmann machine

(Restricted) Boltzmann Machines

Z_θ is analytic but computationally intractable, $2^{|H|+|X|}$ states.

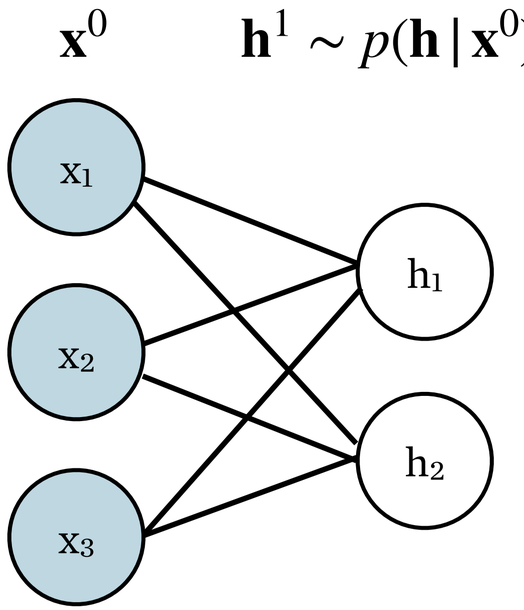


Freund and Haussler [1994]: RBMs are Products of Experts ⇒ still intractable.

Hinton [2002] introduced Contrastive Divergence (CD) to train RBMs.

Restricted Boltzmann Machines

$$\nabla_\theta \Delta \mathrm{KL} \approx \mathbb{E}_{\mathbf{x}\sim p^0(\mathbf{x})}[\frac{dE_\theta(\mathbf{x})}{d\theta}] - \mathbb{E}_{\mathbf{x}\sim p^t_\theta(\mathbf{x})}[\frac{dE_\theta(\mathbf{x})}{d_\theta}]$$

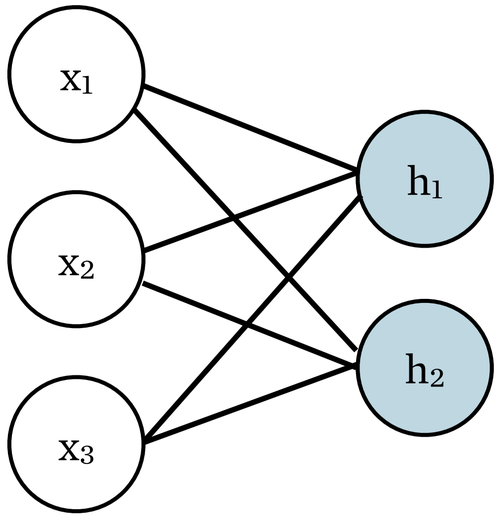


Gibbs sampling in an RBM for contrastive divergence.
RBMs (generally PoEs): easy to sample $p(\mathbf{x}\,|\,\mathbf{h})$ and $p(\mathbf{h}\,|\,\mathbf{x})$.

Restricted Boltzmann Machines

$$\nabla_\theta \Delta \mathrm{KL} \approx \mathbb{E}_{\mathbf{x}\sim p^0(\mathbf{x})}[\frac{dE_\theta(\mathbf{x})}{d\theta}] - \mathbb{E}_{\mathbf{x}\sim p^t_\theta(\mathbf{x})}[\frac{dE_\theta(\mathbf{x})}{d_\theta}]$$

$\mathbf{x}^1 \sim p(\mathbf{x}\,|\,\mathbf{h}^1)$ $\mathbf{h}^1$



Gibbs sampling in an RBM for contrastive divergence.
RBMs (generally PoEs): easy to sample $p(\mathbf{x}\,|\,\mathbf{h})$ and $p(\mathbf{h}\,|\,\mathbf{x})$.

Some results

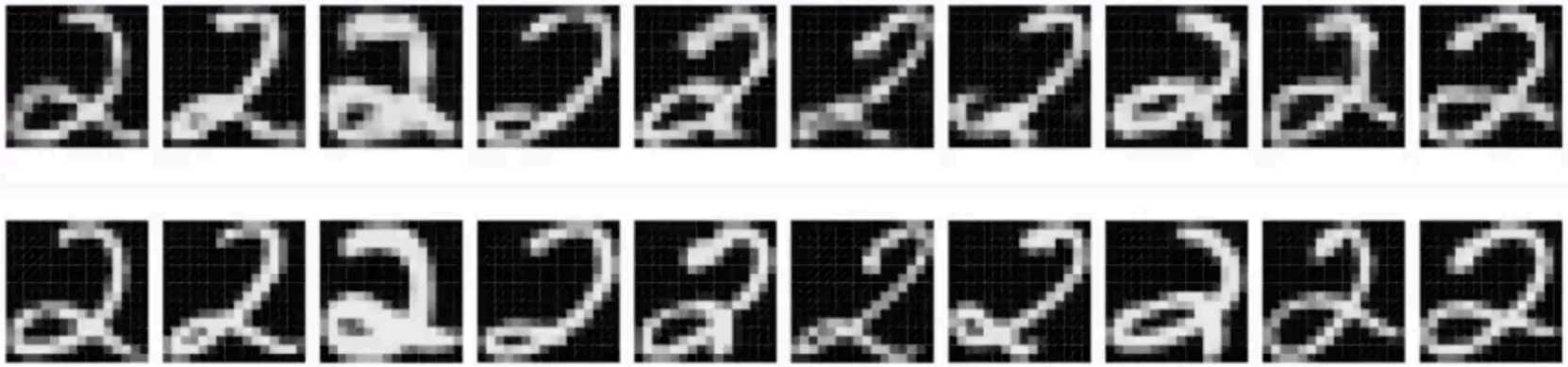


Figure 7: MNIST images (top) and their reconstrutions (bottom) by an RBM.

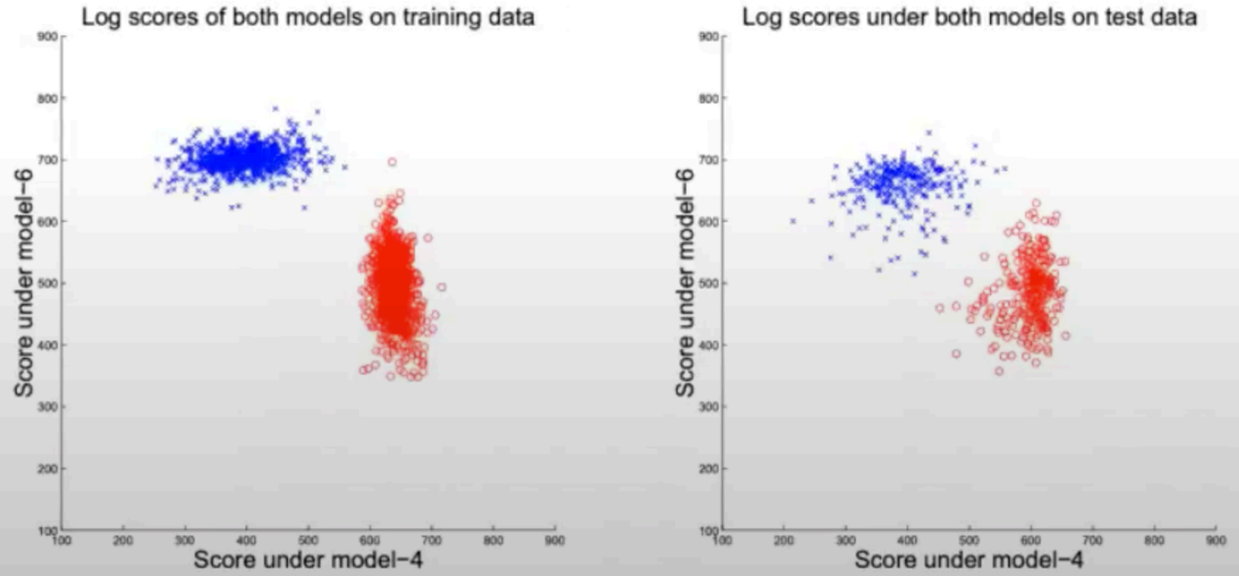
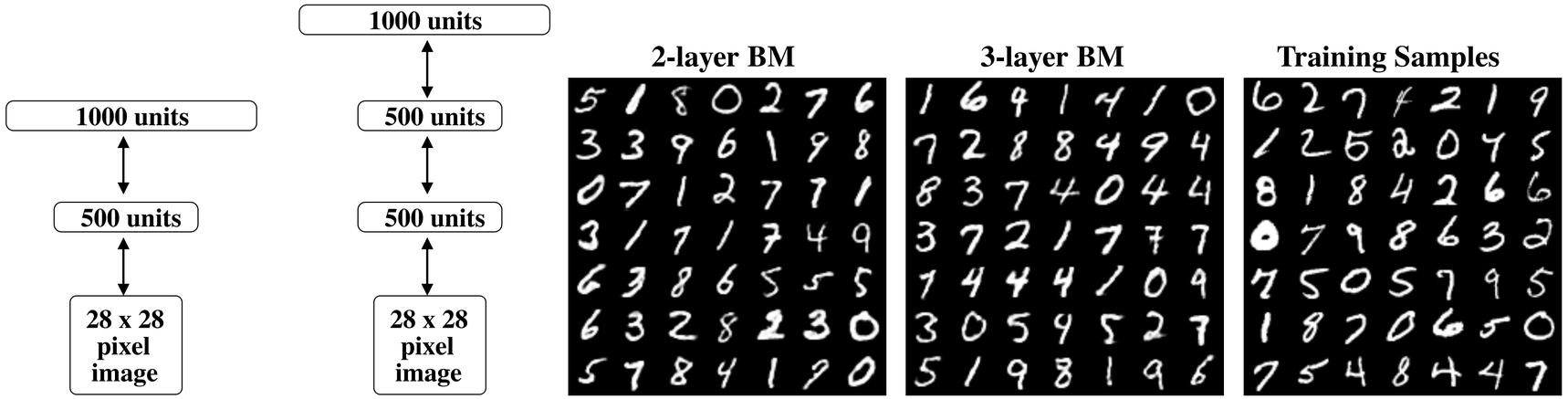
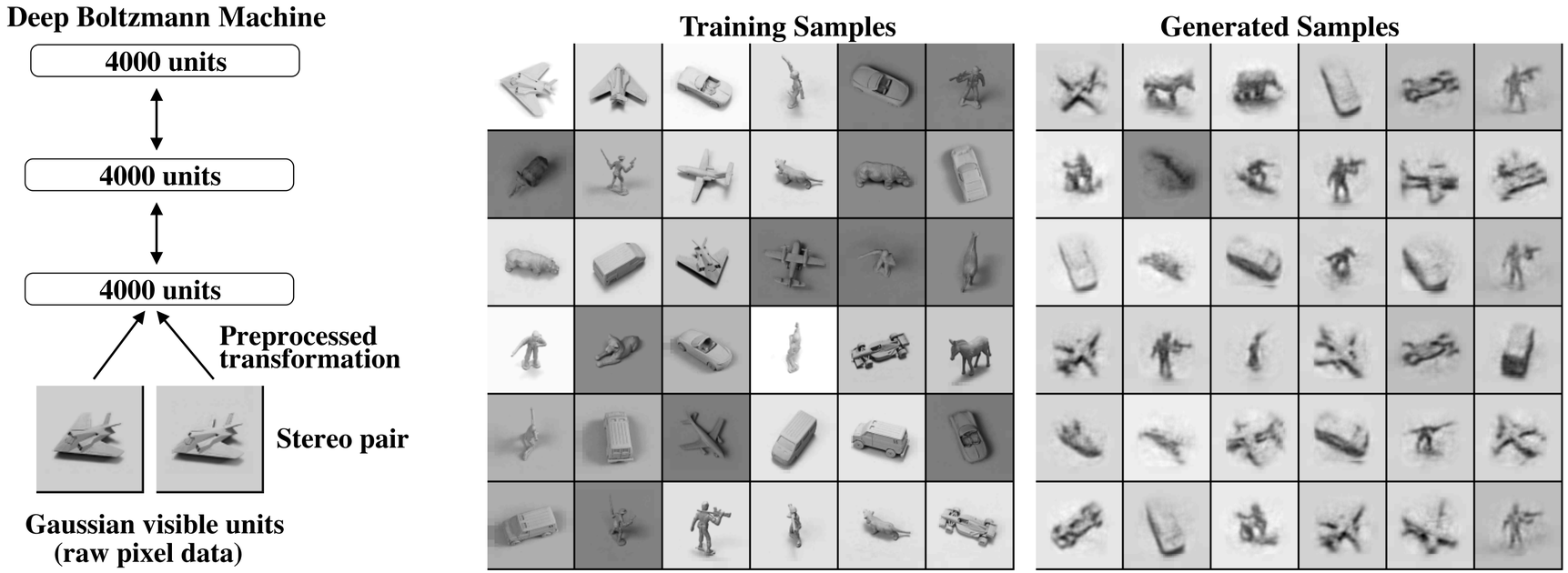


Figure 8: Energies of digits 4 and 6 under RBMs trained with digits 4 and 6.

Deep Boltzmann Machines



Training DBMs on the MNIST data set.



Generating samples after training on the NORB data set.

Salakhutdinov and Hinton, 2008: Deep Boltzmann Machines.

Summary so far

- EBMs are a model class with intractable log-likelihoods (due to Z_θ).
- We can train EBMs by Contrastive Divergence
  1. Set up a Markov chain for p_θ^t
  2. Minimise $\Delta\mathrm{KL} = \mathrm{KL}(p^0 \| p_\theta^\infty) - \mathrm{KL}(p_\theta^t \| p_\theta^\infty)$.
  3. Cancellation of Z_θ makes gradients tractable.
  4. MCMC particularly easy for PoEs (conditional independence).
- Can we leverage recent developments in Deep Learning for EBMs?
- Are there alternatives for training EBMs?

Back to the future

Contrastive Divergence I

We want to maximise the likelihood

$$p_\theta(\mathbf{x}) = \frac{e^{-E_\theta(\mathbf{x})}}{Z_\theta}$$

but we can't compute the normalising constant

$$Z_\theta = \int e^{-E_\theta(\mathbf{x})} d\mathbf{x}.$$

However, it turns out that with a few tricks we can compute the gradient of the log-likelihood

$$\nabla_\theta \log p_\theta(\mathbf{x}) = -\nabla_\theta E_\theta(\mathbf{x}) - \nabla_\theta \log Z_\theta.$$

Contrastive Divergence II

We first term $\nabla_\theta E_\theta(\mathbf{x})$ is easy to compute with sampling. But, the second term $\nabla_\theta \log Z_\theta$ is intractable to compute exactly. It can, however, be approximated, with simple MC as

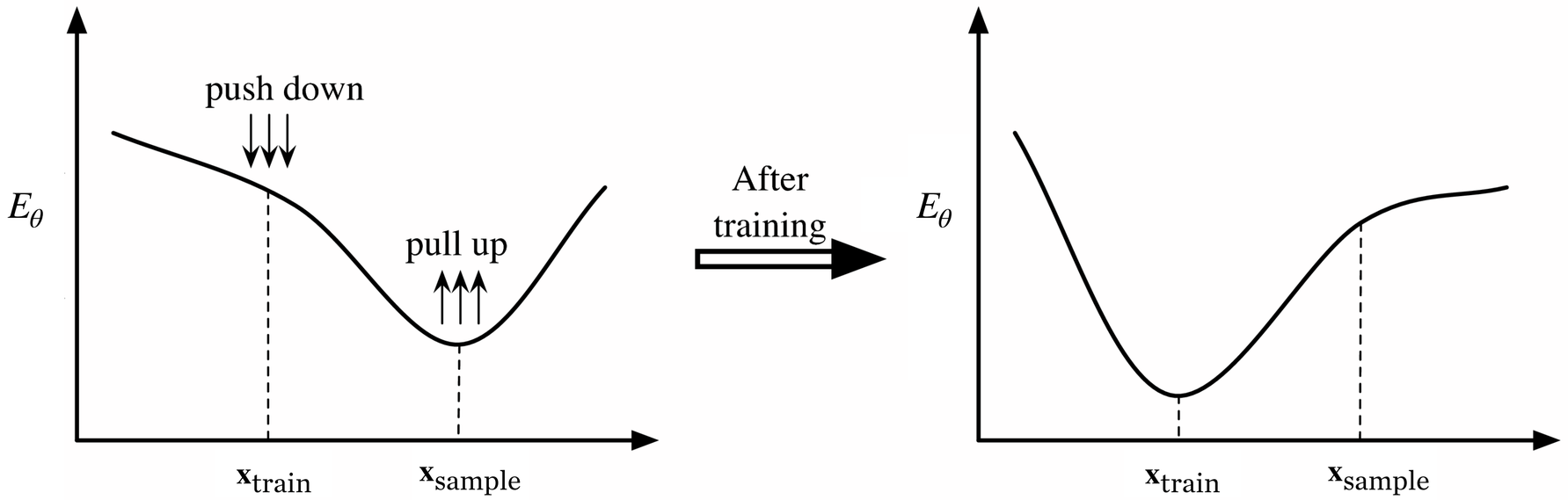
$$\nabla_\theta \log Z_\theta = \mathbb{E}_{\mathbf{x}\sim p_\theta(\mathbf{x})}[-\nabla_\theta E_\theta(\mathbf{x})] \approx \frac{1}{N}\sum_{n}^{N} -\nabla_\theta E_\theta(\mathbf{x}_n), \mathbf{x}_n \sim p_\theta(\mathbf{x}).$$

Thus, we are taking gradient steps in the direction

$$\nabla_\theta E_\theta(\mathbf{x}_{\text{train}}) - \nabla_\theta E_\theta(\mathbf{x}_{\text{sample}}).$$

Contrastive Divergence III

(Yann LeCun et al., 2006: A tutorial on energy-based learning)



Taking steps in the direction of $\nabla_\theta E_\theta(\mathbf{x}_{\text{train}}) - \nabla_\theta E_\theta(\mathbf{x}_{\text{sample}})$

Contrastive Divergence IV

Langevin sampling

Sampling from $p_\theta(\mathbf{x})$ is non-trivial, so that we resort to further approximation.

A common choice is to use Langevin MCMC.

$$\mathbf{x}^{t+1} \leftarrow \mathbf{x}^t + \frac{\epsilon^2}{2}\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}^t) + \epsilon \mathbf{z}^t, \quad t = 0,1,\ldots,T-1$$

where $\mathbf{z}^t \sim \mathcal{N}(0,1)$ and $\mathbf{x}^0 \sim p(\mathbf{x})$.

Note: $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}^t) = -\nabla_{\mathbf{x}} E_\theta(\mathbf{x}) - \cancel{\nabla_{\mathbf{x}} \log Z_\theta} \approx -\nabla_{\mathbf{x}} E_\theta(\mathbf{x})$.

Improved CD for EBMs

The CD gradient doesn't come from the log-likelihood, but rather

$$\Delta \mathrm{KL} = \mathrm{KL}(p^0 \| p_\theta^\infty) - \mathrm{KL}(p_\theta^t \| p_\theta^\infty)$$

Adding the KL term is equivalent to adding a KL loss

$$\mathscr{L}_{\mathrm{KL}} = \underbrace{\mathbb{E}_{p_\theta^t(\mathbf{x})}[\mathbf{x}]}_{(1)} + \underbrace{\mathbb{E}_{p_\theta^t(\mathbf{x})}[\log \mathbf{p}_\theta^{\mathbf{t}}(\mathbf{x})]}_{(2)}$$

Estimation of the KL loss is fairly involved.

(1) requires differentiating through the MCMC sampling

(2) is estimated via nearest-neighbours approximation.

Improved CD for EBMs

$$\nabla_\theta \Delta \mathrm{KL} = \int \left[p^0 \frac{dE_\theta}{d\theta} - p_\theta^t \frac{dE_\theta}{d\theta} - \frac{dp_\theta^t}{d\theta} \frac{\delta}{\delta p_\theta^t} \mathrm{KL}(p_\theta^t \| p_\theta^\infty) \right] d\mathbf{x}$$

The CD gradient doesn't come from the log-likelihood, but rather

$$\Delta \mathrm{KL} = \mathrm{KL}(p^0 \| p_\theta^\infty) - \mathrm{KL}(p_\theta^t \| p_\theta^\infty)$$

Adding the KL term is equivalent to adding a KL loss

$$\mathscr{L}_{\mathrm{KL}} = \underbrace{\mathbb{E}_{p_\theta^t(\mathbf{x})}[\mathbf{x}]}_{(1)} + \underbrace{\mathbb{E}_{p_\theta^t(\mathbf{x})}[\log \mathbf{p}_\theta^\mathbf{t}(\mathbf{x})]}_{(2)}$$

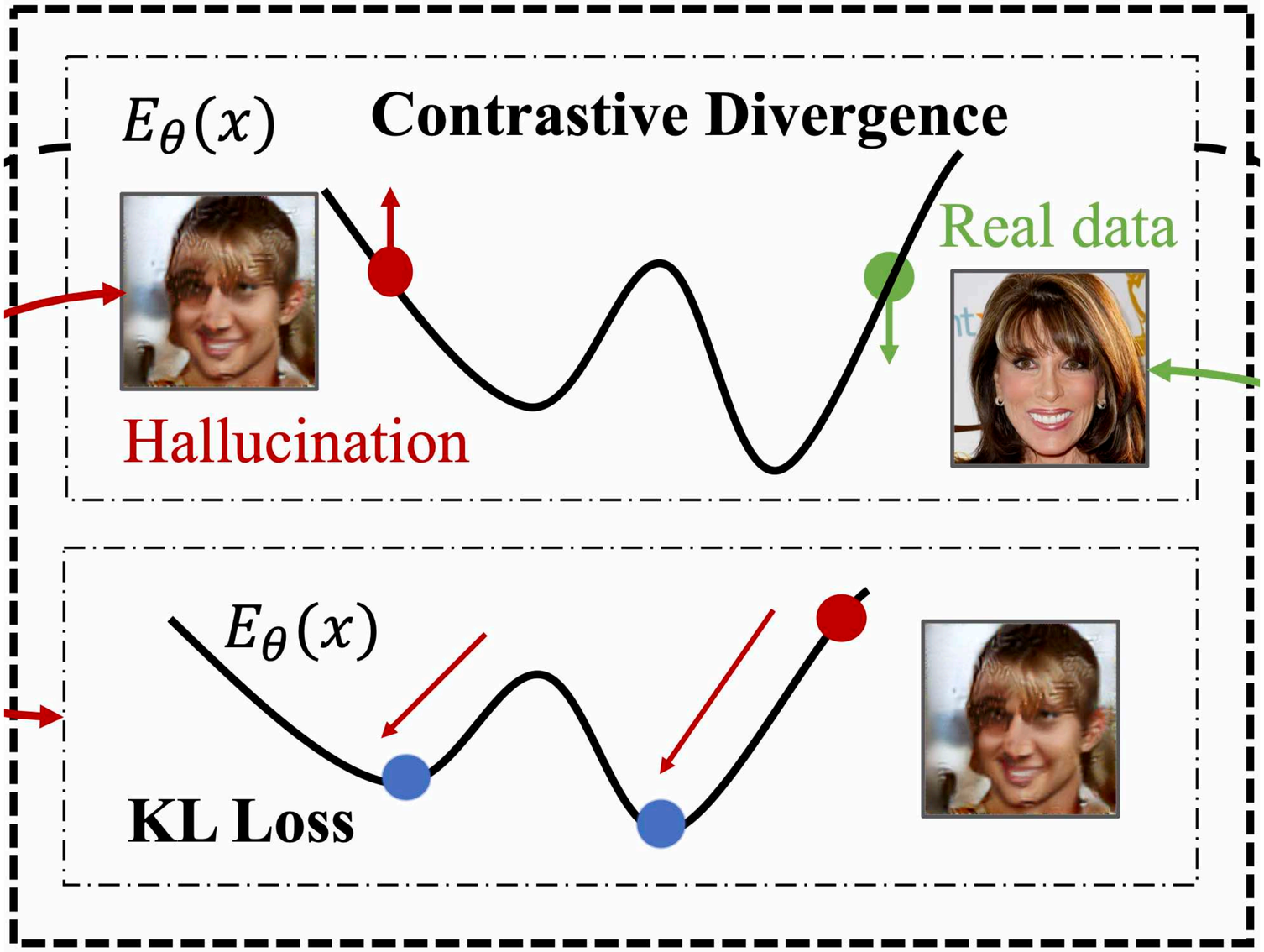
Estimation of the KL loss is fairly involved.

(1) requires differentiating through the MCMC sampling

(2) is estimated via nearest-neighbours approximation.

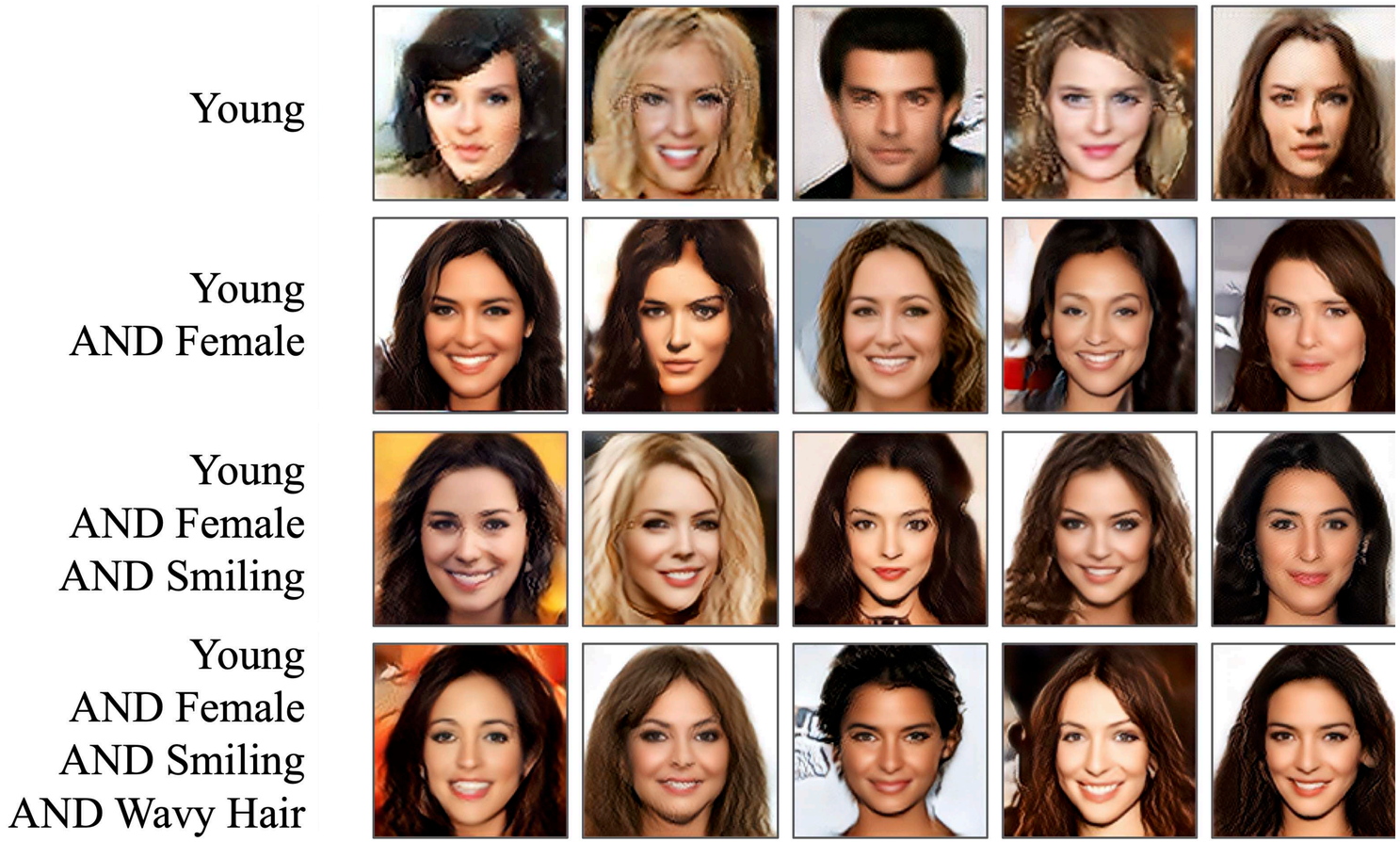
Improved Contrastive Divergence Training for EBMs

Yilun Du et al., 2021



Improved Contrastive Divergence Training for EBMs

Yilun Du et al., 2021



Score Matching

If $f(\mathbf{x})$ and $g(\mathbf{x})$ have equal first derivatives (aka score functions), then $f(\mathbf{x}) \equiv g(\mathbf{x}) + \text{constant}$. When $f(\mathbf{x})$ and $g(\mathbf{x})$ are log PDFs $f(\mathbf{x}) \equiv g(\mathbf{x})$.

Score Matching

If $f(\mathbf{x})$ and $g(\mathbf{x})$ have equal first derivatives (aka score functions), then $f(\mathbf{x}) \equiv g(\mathbf{x}) + \text{constant}$. When $f(\mathbf{x})$ and $g(\mathbf{x})$ are log PDFs $f(\mathbf{x}) \equiv g(\mathbf{x})$.

Hyvärinen and Dayan [2005] propose to learn EBMs by minimising

Score Matching

If $f(\mathbf{x})$ and $g(\mathbf{x})$ have equal first derivatives (aka score functions), then $f(\mathbf{x}) \equiv g(\mathbf{x}) + \text{constant}$. When $f(\mathbf{x})$ and $g(\mathbf{x})$ are log PDFs $f(\mathbf{x}) \equiv g(\mathbf{x})$.

Hyvärinen and Dayan [2005] propose to learn EBMs by minimising

$$D_F(p_{\text{data}}(\mathbf{x}) \| p_\theta(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\frac{1}{2} \| \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) \|^2].$$

Score Matching

If $f(\mathbf{x})$ and $g(\mathbf{x})$ have equal first derivatives (aka score functions), then $f(\mathbf{x}) \equiv g(\mathbf{x}) + \text{constant}$. When $f(\mathbf{x})$ and $g(\mathbf{x})$ are log PDFs $f(\mathbf{x}) \equiv g(\mathbf{x})$.

Hyvärinen and Dayan [2005] propose to learn EBMs by minimising

$$D_F(p_{\text{data}}(\mathbf{x}) \| p_\theta(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\frac{1}{2} \| \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) \|^2].$$

- The expectation can be approximated with a simple MC estimator

Score Matching

If $f(\mathbf{x})$ and $g(\mathbf{x})$ have equal first derivatives (aka score functions), then $f(\mathbf{x}) \equiv g(\mathbf{x}) + \text{constant}$. When $f(\mathbf{x})$ and $g(\mathbf{x})$ are log PDFs $f(\mathbf{x}) \equiv g(\mathbf{x})$.

Hyvärinen and Dayan [2005] propose to learn EBMs by minimising

$$D_F(p_{\text{data}}(\mathbf{x}) \| p_\theta(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\frac{1}{2} \| \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) \|^2].$$

- The expectation can be approximated with a simple MC estimator
- The term $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} \log E_\theta(\mathbf{x})$.

Score Matching

If $f(\mathbf{x})$ and $g(\mathbf{x})$ have equal first derivatives (aka score functions), then $f(\mathbf{x}) \equiv g(\mathbf{x}) + \text{constant}$. When $f(\mathbf{x})$ and $g(\mathbf{x})$ are log PDFs $f(\mathbf{x}) \equiv g(\mathbf{x})$.

Hyvärinen and Dayan [2005] propose to learn EBMs by minimising

$$D_F(p_{\text{data}}(\mathbf{x}) \| p_\theta(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\frac{1}{2} \| \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) \|^2].$$

- The expectation can be approximated with a simple MC estimator
- The term $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} \log E_\theta(\mathbf{x})$.
- Unfortunately, the term $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$ is intractable.

Score Matching

If $f(\mathbf{x})$ and $g(\mathbf{x})$ have equal first derivatives (aka score functions), then $f(\mathbf{x}) \equiv g(\mathbf{x}) + \text{constant}$. When $f(\mathbf{x})$ and $g(\mathbf{x})$ are log PDFs $f(\mathbf{x}) \equiv g(\mathbf{x})$.

Hyvärinen and Dayan [2005] propose to learn EBMs by minimising

$$D_F(p_{\text{data}}(\mathbf{x}) \| p_\theta(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\frac{1}{2} \| \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) \|^2].$$

- The expectation can be approximated with a simple MC estimator
- The term $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} \log E_\theta(\mathbf{x})$.
- Unfortunately, the term $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$ is intractable.

However, using integration by parts, we can rewrite this as

Score Matching

If $f(\mathbf{x})$ and $g(\mathbf{x})$ have equal first derivatives (aka score functions), then $f(\mathbf{x}) \equiv g(\mathbf{x}) + \text{constant}$. When $f(\mathbf{x})$ and $g(\mathbf{x})$ are log PDFs $f(\mathbf{x}) \equiv g(\mathbf{x})$.

Hyvärinen and Dayan [2005] propose to learn EBMs by minimising

$$D_F(p_{\text{data}}(\mathbf{x}) \| p_\theta(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\frac{1}{2} \| \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) \|^2].$$

- The expectation can be approximated with a simple MC estimator
- The term $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} \log E_\theta(\mathbf{x})$.
- Unfortunately, the term $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$ is intractable.

However, using integration by parts, we can rewrite this as

$$D_F(p_{\text{data}}(\mathbf{x}) \| p_\theta(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\frac{1}{2} \sum_{i=1}^{d} \left(\frac{\partial E_\theta(\mathbf{x})}{\partial x_i} \right)^2 + \frac{\partial^2 E_\theta(\mathbf{x})}{(\partial x_i)^2} \right] + c.$$

Denoising score matching

Naive score matching has two potential problematic requirements:

1. $p_{\text{data}}(\mathbf{x})$ is continuously differentiable and finite everywhere, and
2. The computation of expensive second-order gradients

Denoising score matching

Naive score matching has two potential problematic requirements:

1. $p_{\text{data}}(\mathbf{x})$ is continuously differentiable and finite everywhere, and
2. The computation of expensive second-order gradients

Problem 1 can be solved by adding noise to each data point $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$,

Denoising score matching

Naive score matching has two potential problematic requirements:

1. $p_{\text{data}}(\mathbf{x})$ is continuously differentiable and finite everywhere, and
2. The computation of expensive second-order gradients

Problem 1 can be solved by adding noise to each data point $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$,

resulting in a noisy data distribution $q(\tilde{\mathbf{x}}) = \int q(\tilde{\mathbf{x}} \mid \mathbf{x})\, p_{\text{data}}(\mathbf{x}) d\mathbf{x}$.

Denoising score matching

Naive score matching has two potential problematic requirements:

1. $p_{\text{data}}(\mathbf{x})$ is continuously differentiable and finite everywhere, and
2. The computation of expensive second-order gradients

Problem 1 can be solved by adding noise to each data point $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$,

resulting in a noisy data distribution $q(\tilde{\mathbf{x}}) = \int q(\tilde{\mathbf{x}} \mid \mathbf{x})\, p_{\text{data}}(\mathbf{x}) d\mathbf{x}$.

Vincent [2011] solve problem 2 by showing:

Denoising score matching

Naive score matching has two potential problematic requirements:

1. $p_{\text{data}}(\mathbf{x})$ is continuously differentiable and finite everywhere, and
2. The computation of expensive second-order gradients

Problem 1 can be solved by adding noise to each data point $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$,

resulting in a noisy data distribution $q(\tilde{\mathbf{x}}) = \int q(\tilde{\mathbf{x}} \,|\, \mathbf{x})\, p_{\text{data}}(\mathbf{x}) d\mathbf{x}$.

Vincent [2011] solve problem 2 by showing:

$$
\begin{aligned}
D_F(q(\tilde{\mathbf{x}}) \| p_\theta(\tilde{\mathbf{x}})) &= \mathbb{E}_{q(\tilde{\mathbf{x}})} \left[\frac{1}{2} \| \nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}}) - \nabla_{\mathbf{x}} \log p_\theta(\tilde{\mathbf{x}}) \|_2^2 \right] \\
&= \mathbb{E}_{q(\mathbf{x}, \tilde{\mathbf{x}})} \left[\frac{1}{2} \| \nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}} \,|\, \mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\tilde{\mathbf{x}}) \|_2^2 \right] + c
\end{aligned}
$$

Denoising score matching

Naive score matching has two potential problematic requirements:

1. $p_{\text{data}}(\mathbf{x})$ is continuously differentiable and finite everywhere, and
2. The computation of expensive second-order gradients

Problem 1 can be solved by adding noise to each data point $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$,

resulting in a noisy data distribution $q(\tilde{\mathbf{x}}) = \int q(\tilde{\mathbf{x}} \mid \mathbf{x})\, p_{\text{data}}(\mathbf{x}) d\mathbf{x}$.

Vincent [2011] solve problem 2 by showing:

$$
\begin{aligned}
D_F(q(\tilde{\mathbf{x}}) \| p_\theta(\tilde{\mathbf{x}})) &= \mathbb{E}_{q(\tilde{\mathbf{x}})} \left[\frac{1}{2} \| \nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}}) - \nabla_{\mathbf{x}} \log p_\theta(\tilde{\mathbf{x}}) \|_2^2 \right] \\
&= \mathbb{E}_{q(\mathbf{x}, \tilde{\mathbf{x}})} \left[\frac{1}{2} \| \nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}} \mid \mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\tilde{\mathbf{x}}) \|_2^2 \right] + c
\end{aligned}
$$

Thus avoiding any expensive second order gradients.

Denoising score matching

Naive score matching has two potential problematic requirements:

1. $p_{\text{data}}(\mathbf{x})$ is continuously differentiable and finite everywhere, and
2. The computation of expensive second-order gradients

Problem 1 can be solved by adding noise to each data point $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$,

resulting in a noisy data distribution $q(\tilde{\mathbf{x}}) = \int q(\tilde{\mathbf{x}} \mid \mathbf{x})\, p_{\text{data}}(\mathbf{x}) d\mathbf{x}$.

Vincent [2011] solve problem 2 by showing:

$$
\begin{aligned}
D_F(q(\tilde{\mathbf{x}}) \| p_\theta(\tilde{\mathbf{x}})) &= \mathbb{E}_{q(\tilde{\mathbf{x}})} \left[\frac{1}{2} \| \nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}}) - \nabla_{\mathbf{x}} \log p_\theta(\tilde{\mathbf{x}}) \|_2^2 \right] \\
&= \mathbb{E}_{q(\mathbf{x}, \tilde{\mathbf{x}})} \left[\frac{1}{2} \| \nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}} \mid \mathbf{x}) - \nabla_{\mathbf{x}} \log p_\theta(\tilde{\mathbf{x}}) \|_2^2 \right] + c
\end{aligned}
$$

Thus avoiding any expensive second order gradients.

New problems: trade-off between estimator variance and noise magnitude. Inconsistency.

Sliced score matching

Sliced score matching minimises the sliced Fisher divergence

$$D_{\mathrm{SF}}(p_{\mathrm{data}} \| p_\theta) = \mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})} \mathbb{E}_{p(\mathbf{v})} \left[\frac{1}{2} (\mathbf{v}^\top \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) - \mathbf{v}^\top \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}))^2 \right]$$

where $p(\mathbf{v})$ denotes a projection distribution such that $E_{p(\mathbf{v})}[\mathbf{v}\mathbf{v}^\top]$ is pos. def.

As before, we can use the chain rule to avoid $\nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x})$:

$$\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})} \mathbb{E}_{p(\mathbf{v})} \left[\frac{1}{2} \sum_{i=1}^{d} \left(\frac{\partial E_\theta(\mathbf{x})}{\partial x_i} v_i \right)^2 + \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial^2 E_\theta(\mathbf{x})}{\partial x_i \partial x_j} v_i v_j \right].$$

However, unlike before, this for has a computational complexity of $\mathcal{O}(d)$:

$$\sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial^2 E_\theta(\mathbf{x})}{\partial x_i \partial x_j} v_i v_j = \sum_{i=1}^{d} \frac{\partial}{\partial x_i} \underbrace{\left(\sum_{j=1}^{d} \frac{\partial E_\theta(\mathbf{x})}{\partial x_j} v_j \right)}_{:= f(x)} v_i$$

Noise Contrastive Estimation

Another alternative training method (Gutmann and Hyvärinen, 2020):

Define a known and tractable reference distribution $p_r(\mathbf{r})$.

Treat Z_θ as a training variable.

$$\log p_\theta = \log \tilde{p}_\theta(\mathbf{x}) + C$$

Draw $\mathbf{x}$ from $p_r(\mathbf{x})$ or from $p_D(\mathbf{x})$ (denoted $y = 0,1$ respectively)

Use EBM to set up a classifier which distinguishes $y = 0,1$

$$p(y = 0 \,|\, \mathbf{x}, \theta) = \frac{p_r(\mathbf{x})}{p_\theta(\mathbf{x}) + p_r(\mathbf{x})}, \quad p(y = 1 \,|\, \mathbf{x}, \theta) = \frac{p_\theta(\mathbf{x})}{p_\theta(\mathbf{x}) + p_r(\mathbf{x})}$$

Noise Contrastive Estimation

Drawn $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim p_r(\mathbf{x})$ and $\mathbf{x}_{N+1}, \ldots, \mathbf{x}_{2N} \sim p_D(\mathbf{x})$. Minimise

$$\mathscr{L}_{\text{NCE}} = \sum_{n=1}^{N} \log p(y_n = 1 \,|\, \mathbf{x}_n, \theta) + \log p(y_{N+n} = 0 \,|\, \mathbf{x}_{N+n}, \theta),$$

binary cross entropy loss for classifying samples.

If there exists θ^* such that $p_{\theta^*} = p_D$, then in the limit $N \to \infty$ we have $\theta \to \theta^*$. Further, θ^* is a unique global optimum.

Observation: Objective automatically takes care of Z_θ.

$$p(y = 0 \,|\, \mathbf{x}, \theta) = \frac{p_r(\mathbf{x})}{p_\theta(\mathbf{x}) + p_r(\mathbf{x})}, \quad p(y = 1 \,|\, \mathbf{x}, \theta) = \frac{p_\theta(\mathbf{x})}{p_\theta(\mathbf{x}) + p_r(\mathbf{x})}$$

Intuition: Trainable variable C cannot go to either $-\infty$ or to ∞.

Conditional Noise Contrastive Estimation

Challenge with NCE: How to choose the reference distribution $p_r(\mathbf{x})$?

Conditional Noise Contrastive Estimation

Challenge with NCE: How to choose the reference distribution $p_r(\mathbf{x})$?

Ceylan and Gutmann [2018] propose $p_r(\mathbf{x}) = \int q(\mathbf{x} \mid \mathbf{x}') p_D(\mathbf{x}') d\mathbf{x}'$ and

Conditional Noise Contrastive Estimation

Challenge with NCE: How to choose the reference distribution $p_r(\mathbf{x})$?

Ceylan and Gutmann [2018] propose $p_r(\mathbf{x}) = \int q(\mathbf{x} \mid \mathbf{x}') p_D(\mathbf{x}') d\mathbf{x}'$ and

$$D(\mathbf{x}, \mathbf{x}') = p(y = 1 \mid \mathbf{x}, \mathbf{x}', \theta) = \frac{q(\mathbf{x}' \mid \mathbf{x}) p_\theta(\mathbf{x})}{q(\mathbf{x}' \mid \mathbf{x})\, p_\theta(\mathbf{x}) + q(\mathbf{x} \mid \mathbf{x}')\, p_\theta(\mathbf{x}')},$$

Conditional Noise Contrastive Estimation

Challenge with NCE: How to choose the reference distribution $p_r(\mathbf{x})$?

Ceylan and Gutmann [2018] propose $p_r(\mathbf{x}) = \int q(\mathbf{x} \mid \mathbf{x}') p_D(\mathbf{x}') d\mathbf{x}'$ and

$$D(\mathbf{x}, \mathbf{x}') = p(y = 1 \mid \mathbf{x}, \mathbf{x}', \theta) = \frac{q(\mathbf{x}' \mid \mathbf{x}) p_\theta(\mathbf{x})}{q(\mathbf{x}' \mid \mathbf{x})\, p_\theta(\mathbf{x}) + q(\mathbf{x} \mid \mathbf{x}')\, p_\theta(\mathbf{x}')},$$

Where a typical choice for q is

Conditional Noise Contrastive Estimation

Challenge with NCE: How to choose the reference distribution $p_r(\mathbf{x})$?

Ceylan and Gutmann [2018] propose $p_r(\mathbf{x}) = \int q(\mathbf{x} \mid \mathbf{x}') p_D(\mathbf{x}') d\mathbf{x}'$ and

$$D(\mathbf{x}, \mathbf{x}') = p(y = 1 \mid \mathbf{x}, \mathbf{x}', \theta) = \frac{q(\mathbf{x}' \mid \mathbf{x}) p_\theta(\mathbf{x})}{q(\mathbf{x}' \mid \mathbf{x})\, p_\theta(\mathbf{x}) + q(\mathbf{x} \mid \mathbf{x}')\, p_\theta(\mathbf{x}')},$$

Where a typical choice for q is

$$q(\mathbf{x}' \mid \mathbf{x}) = \mathcal{N}(\mathbf{x}'; \mathbf{x}, \sigma^2 \mathbf{I})$$

Conditional Noise Contrastive Estimation

Challenge with NCE: How to choose the reference distribution $p_r(\mathbf{x})$?

Ceylan and Gutmann [2018] propose $p_r(\mathbf{x}) = \int q(\mathbf{x} \mid \mathbf{x}') p_D(\mathbf{x}') d\mathbf{x}'$ and

$$D(\mathbf{x}, \mathbf{x}') = p(y = 1 \mid \mathbf{x}, \mathbf{x}', \theta) = \frac{q(\mathbf{x}' \mid \mathbf{x}) p_\theta(\mathbf{x})}{q(\mathbf{x}' \mid \mathbf{x})\, p_\theta(\mathbf{x}) + q(\mathbf{x} \mid \mathbf{x}')\, p_\theta(\mathbf{x}')},$$

Where a typical choice for q is

$$q(\mathbf{x}' \mid \mathbf{x}) = \mathcal{N}(\mathbf{x}'; \mathbf{x}, \sigma^2 \mathbf{I})$$

Observation: Above equation for D is invariant to scaling p_θ, and Z_θ cancels.

Conditional Noise Contrastive Estimation

Challenge with NCE: How to choose the reference distribution $p_r(\mathbf{x})$?

Ceylan and Gutmann [2018] propose $p_r(\mathbf{x}) = \int q(\mathbf{x} \mid \mathbf{x}') p_D(\mathbf{x}') d\mathbf{x}'$ and

$$D(\mathbf{x}, \mathbf{x}') = p(y = 1 \mid \mathbf{x}, \mathbf{x}', \theta) = \frac{q(\mathbf{x}' \mid \mathbf{x}) p_\theta(\mathbf{x})}{q(\mathbf{x}' \mid \mathbf{x})\, p_\theta(\mathbf{x}) + q(\mathbf{x} \mid \mathbf{x}')\, p_\theta(\mathbf{x}')},$$

Where a typical choice for q is

$$q(\mathbf{x}' \mid \mathbf{x}) = \mathcal{N}(\mathbf{x}'; \mathbf{x}, \sigma^2 \mathbf{I})$$

Observation: Above equation for D is invariant to scaling p_θ, and Z_θ cancels.

Similarly to NCE, set up classification task and minimise

Conditional Noise Contrastive Estimation

Challenge with NCE: How to choose the reference distribution $p_r(\mathbf{x})$?

Ceylan and Gutmann [2018] propose $p_r(\mathbf{x}) = \int q(\mathbf{x} \mid \mathbf{x}') p_D(\mathbf{x}') d\mathbf{x}'$ and

$$D(\mathbf{x}, \mathbf{x}') = p(y = 1 \mid \mathbf{x}, \mathbf{x}', \theta) = \frac{q(\mathbf{x}' \mid \mathbf{x}) p_\theta(\mathbf{x})}{q(\mathbf{x}' \mid \mathbf{x})\, p_\theta(\mathbf{x}) + q(\mathbf{x} \mid \mathbf{x}')\, p_\theta(\mathbf{x}')},$$

Where a typical choice for q is

$$q(\mathbf{x}' \mid \mathbf{x}) = \mathcal{N}(\mathbf{x}'; \mathbf{x}, \sigma^2 \mathbf{I})$$

Observation: Above equation for D is invariant to scaling p_θ, and Z_θ cancels.

Similarly to NCE, set up classification task and minimise

$$\mathscr{L}_{\text{CNCE}} = -\sum_{n=1}^{N} [\log D(\mathbf{x}_n, \mathbf{x}'_n) + \log(1 - D(\mathbf{x}'_n, \mathbf{x}_n))]$$

Conditional Noise Contrastive Estimation

Challenge with NCE: How to choose the reference distribution $p_r(\mathbf{x})$?

Ceylan and Gutmann [2018] propose $p_r(\mathbf{x}) = \int q(\mathbf{x} \mid \mathbf{x}') p_D(\mathbf{x}') d\mathbf{x}'$ and

$$D(\mathbf{x}, \mathbf{x}') = p(y = 1 \mid \mathbf{x}, \mathbf{x}', \theta) = \frac{q(\mathbf{x}' \mid \mathbf{x}) p_\theta(\mathbf{x})}{q(\mathbf{x}' \mid \mathbf{x})\, p_\theta(\mathbf{x}) + q(\mathbf{x} \mid \mathbf{x}')\, p_\theta(\mathbf{x}')},$$

Where a typical choice for q is

$$q(\mathbf{x}' \mid \mathbf{x}) = \mathcal{N}(\mathbf{x}'; \mathbf{x}, \sigma^2 \mathbf{I})$$

Observation: Above equation for D is invariant to scaling p_θ, and Z_θ cancels.

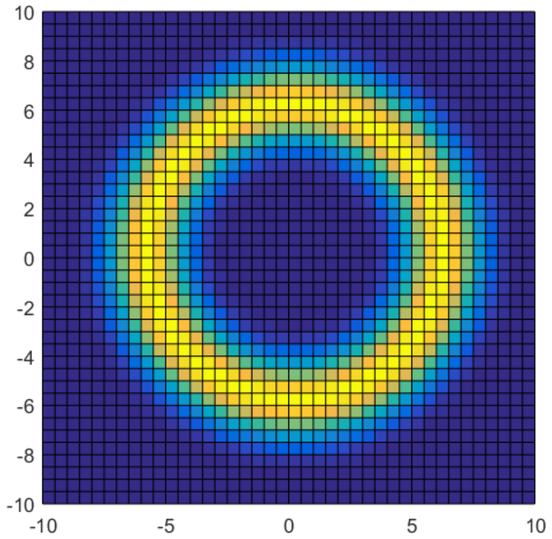
Similarly to NCE, set up classification task and minimise

$$\mathscr{L}_{\text{CNCE}} = -\sum_{n=1}^{N} [\log D(\mathbf{x}_n, \mathbf{x}'_n) + \log(1 - D(\mathbf{x}'_n, \mathbf{x}_n))]$$

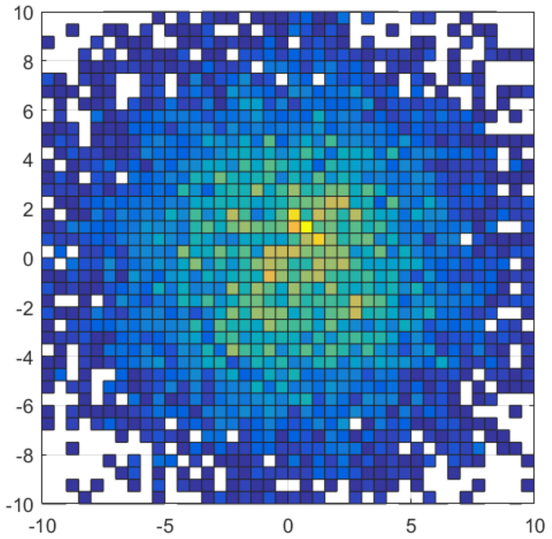
where $\mathbf{x}_1, \ldots, \mathbf{x}_N \sim p_D(\mathbf{x})$ and $\mathbf{x}'_1, \ldots, \mathbf{x}'_N \sim q(\mathbf{x}' \mid \mathbf{x})$.

Conditional Noise Contrastive Estimation

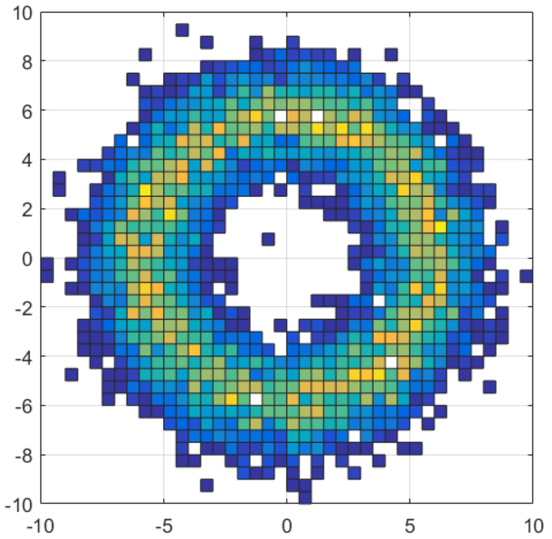
Conditional Noise-Contrastive Estimation of Unnormalised Models
Ceylan, Gutmann, 2018



(a) Contour plot of the data pdf



(b) NCE noise (histogram)



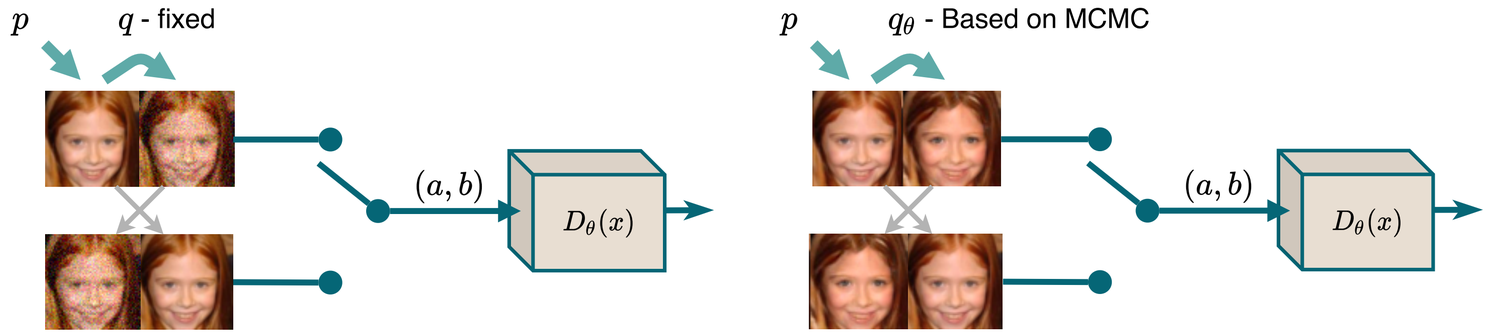
(c) CNCE noise (histogram)

Intuition: Contrastive data closer to real data so classification is more challenging, training stays informative for longer.

By comparison, NCE classification is much easier. Classification task solved quickly, at which point the EBM stops learning.

Relation between (C)NCE and CD

Yair and Michaeli (2021) Contrastive Divergence Learning is a Time Reversal Adversarial Game



(a) CNCE (b) CD-1

The CNCE update rule can be written as

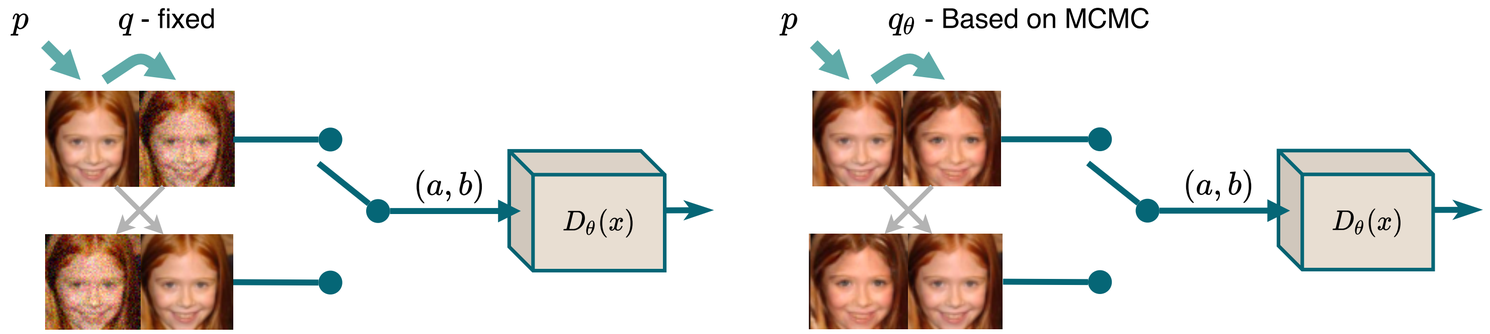
$$\Delta\theta_{\text{CNCE}} = -\nabla_\theta \mathscr{L}_{\text{CNCE}} = \sum_{n=1}^{N} (1 - D(\mathbf{x}_n, \mathbf{x}'_n))[\nabla_\theta \log p_\theta(\mathbf{x}_n) - \nabla_\theta \log p_\theta(\mathbf{x}'_n)]$$

Intuition:

1. Term $\nabla_\theta \log p_\theta(\mathbf{x}_n)$ encourages high probability near the data
2. Term $\nabla_\theta \log p_\theta(\mathbf{x}'_n)$ encourages low probability at contrastive samples
3. Term $1 - D(\mathbf{x}_n, \mathbf{x}'_n)$ downweighs easily-classified pairs.
4. CNCE keeps $D(\mathbf{x}_n, \mathbf{x}'_n)$ close to 1/2 for longer, compared to NCE.

Relation between (C)NCE and CD

Yair and Michaeli (2021) Contrastive Divergence Learning is a Time Reversal Adversarial Game



(a) CNCE (b) CD-1

$$\Delta\theta_{\mathrm{CNCE}} = \sum_{n=1}^{N} (1 - D(\mathbf{x}_n, \mathbf{x}'_n))[\nabla_\theta \log p_\theta(\mathbf{x}_n) - \nabla_\theta \log p_\theta(\mathbf{x}'_n)]$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_N \sim p_D(\mathbf{x})$ and $\mathbf{x}'_1, \ldots, \mathbf{x}'_N \sim p_r(\mathbf{x})$.

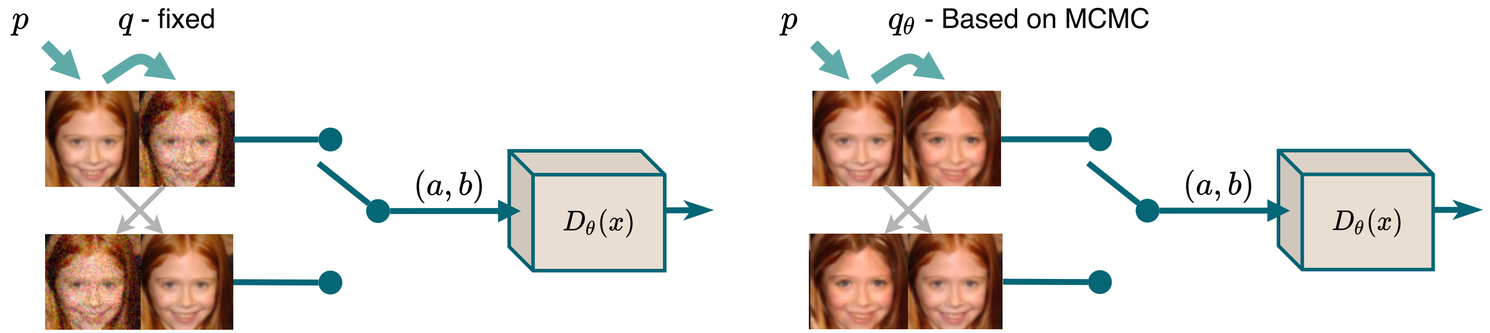
Relation to CD: Let $q(\mathbf{x}' \mid \mathbf{x}) = q_\theta(\mathbf{x}' \mid \mathbf{x})$ be a reversible Markov chain, with stationary distribution p_θ.

The optimal classifier becomes a random guess

$$D(\mathbf{x}_n, \mathbf{x}'_n) = \frac{q(\mathbf{x}' \mid \mathbf{x}) p(\mathbf{x})}{q(\mathbf{x}' \mid \mathbf{x}) p_\theta(\mathbf{x}) + q(\mathbf{x} \mid \mathbf{x}') p_\theta(\mathbf{x}')} = \frac{1}{2}$$

Relation between (C)NCE and CD

Yair and Michaeli (2021) Contrastive Divergence Learning is a Time Reversal Adversarial Game



(a) CNCE (b) CD-1

Under this model, the update rule becomes identical to CD

$$\Delta\theta_{\text{CNCE}} = \frac{1}{2}\sum_{n=1}^{N}[\nabla_\theta \log p_\theta(\mathbf{x}_n) - \nabla_\theta \log p_\theta(\mathbf{x}'_n)]$$

Detail:

Sample $\mathbf{x}'_n \sim q_\theta(\mathbf{x}'|\mathbf{x})$ depends on θ. It is necessary to stop gradients through q_θ for equivalence between CD and CNCE. Stopping gradients is equivalent to ignoring the intractable CD term.

Example applications

Combining VAE with EBM

VAEBM: A Symbiosis between Variational Autoencoders and Energy-based Models
Zhisheng Xiao et al., 2020

ψ, θ are trained by maximising the marginal log-likelihood, in 2 steps:

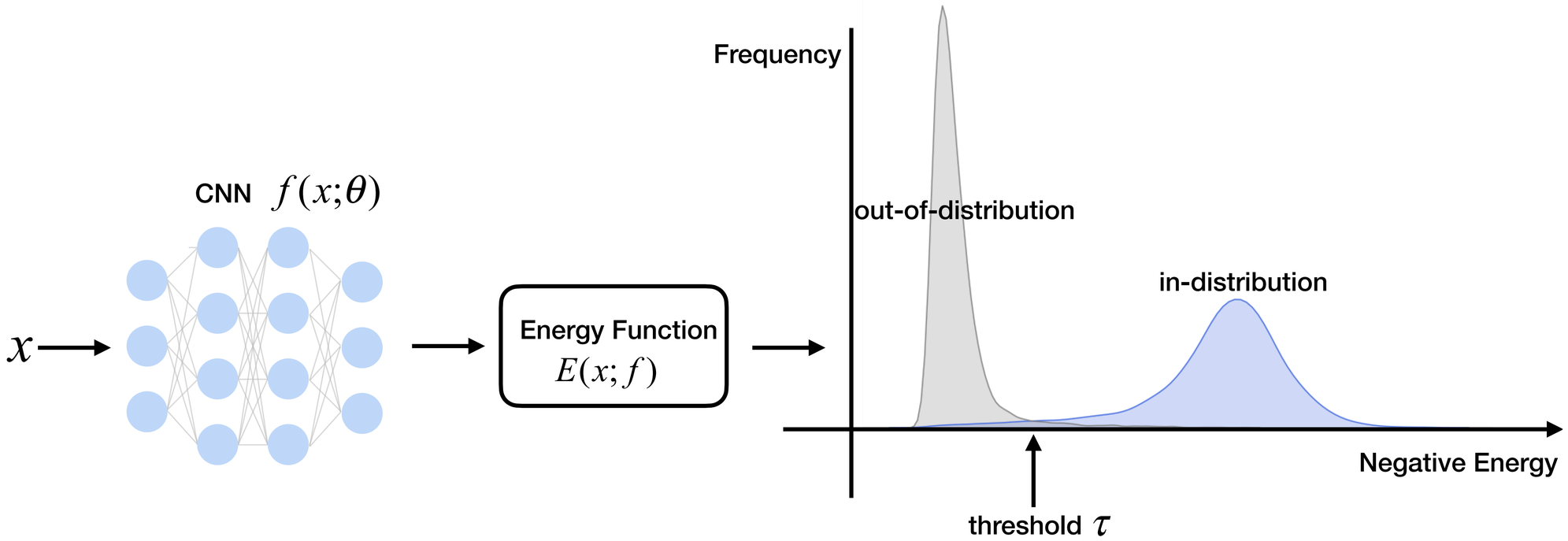
$$\begin{aligned}\log h_{\psi,\theta}(\mathbf{x}) = \log p_\theta(\mathbf{x}) &= E_\psi(\mathbf{x}) - \log Z_{\psi,\theta} \\ &\geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}\,|\,\mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z}\,|\,\mathbf{x})\|p(\mathbf{z}))}_{\mathscr{L}_{\mathrm{VAE}}(\mathbf{x},\theta,\phi)} \underbrace{-E_\psi(\mathbf{x}) - \log Z_{\psi,\theta}}_{\mathscr{L}_{\mathrm{EBM}}(\mathbf{x},\psi,\theta)}\end{aligned}$$

Step 1: train the VAE via $\mathscr{L}_{\mathrm{VAE}}$.

Step 2: Fix the VAE, train the EBM via $\mathscr{L}_{\mathrm{EBM}}$ with CD.

Energy-based OOD detection

Energy-based Out-of-distribution Detection
Waiting Liu et al, 2021



Either apply the energy score to pre-trained NNs, or use it as an additional loss during training.

Summary

Energy-based models

- Energy-based models are a flexible class of models, expanding our modelling toolbox
- Several approaches to training EBMs, each with + and -
    1. Contrastive divergence (CD)
    2. Score matching (SM), denoising SM, sliced SM.
    3. Noise Contrastive Estimation (NCE), and conditional NCE (CNCE).
- Some of these are equivalent under certain conditions.

Training Method	Fast	Stable training	High dimensions	No aux. model	Unrestricted architecture	Approximates likelihood
Markov chain Monte Carlo	✗	✗	✓	✓	✓	✓
Score Matching Approaches	✓	✗	✓	✓	✗	✗
Noise Contrastive Approaches	✓	✓	✗	✗	✓	✗

(Adapted from [Grathwohl et al., 2020a].)