Energy Based Models Reading Group

James Allingham, Stratis Markou

1st December 2020

UNIVERSITY OF CAMBRIDGE

Introduction

What are energy based models?

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

$$p_\theta(\boldsymbol{x}) = \frac{\tilde{p}_\theta(\boldsymbol{x})}{Z_\theta}. \tag{1}$$

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

$$E_\theta(\boldsymbol{x}) = -\log \tilde{p}_\theta(\boldsymbol{x}). \tag{2}$$

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

$$p_\theta(\boldsymbol{x}) = \frac{e^{-E_\theta(\boldsymbol{x})}}{Z_\theta} \implies E_\theta(\boldsymbol{x}) = -\log p_\theta(\boldsymbol{x}) - \log Z_\theta, \tag{3}$$

Energy Based Models (EBMs)

We use the term Energy Based Model (EBM) for energy functions $E_\theta(\boldsymbol{x})$ where $Z_\theta = \int e^{-E_\theta(\boldsymbol{x})} d\boldsymbol{x}$ 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. ARMs, VAEs, NFs).

  We just need an $E_\theta : \mathcal{X} \to \mathbb{R}$, such that Z_θ is finite.

  Not constrained to models with tractable likelihoods.

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

Example EBM Applications

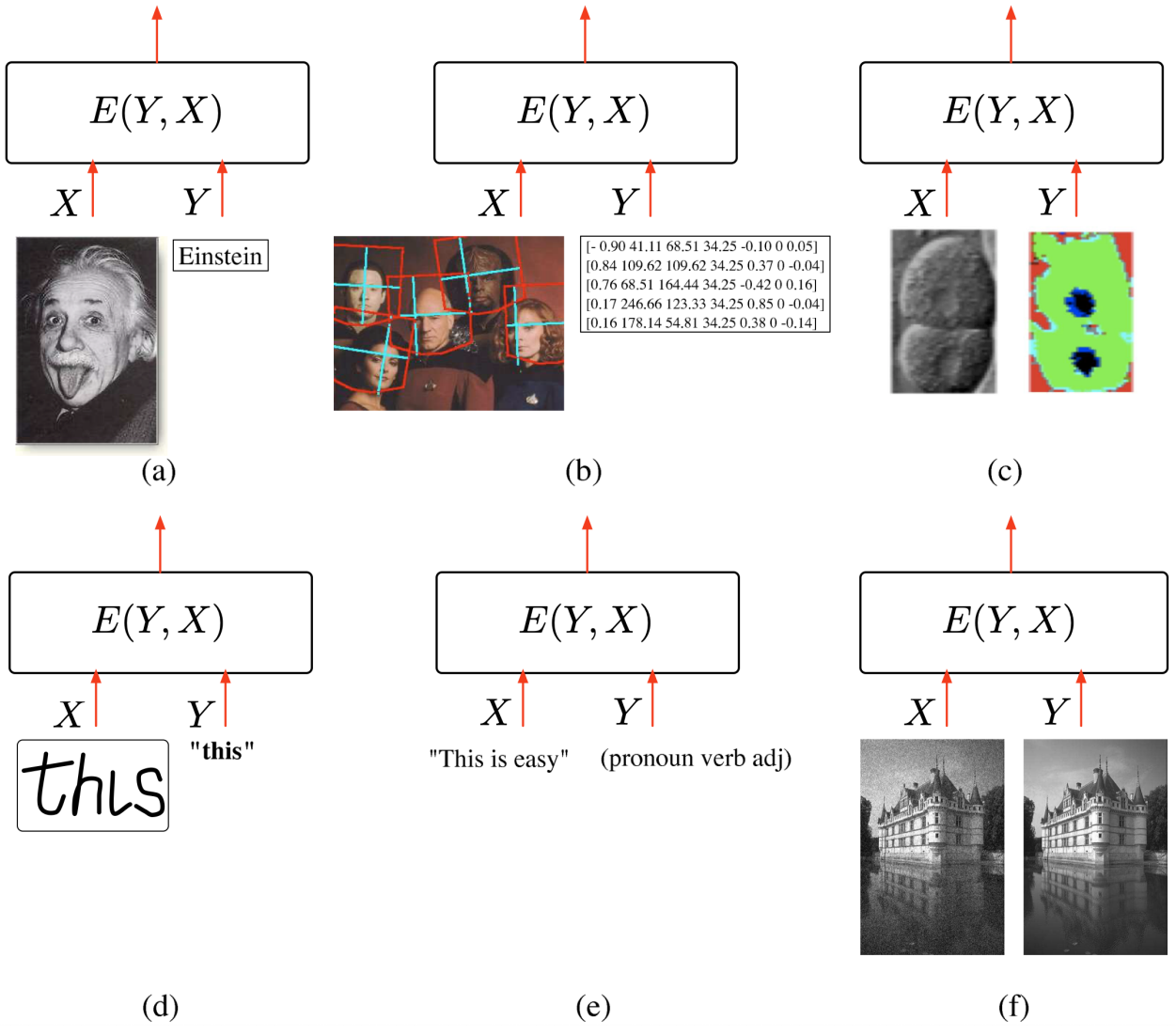


Figure 1: Examples of EBM applications [LeCun et al., 2006].

Old-school

An Example – Product of Experts [Hinton, 2002]

Product of N experts has the form

$$p_\theta(\boldsymbol{x}) = \frac{1}{Z_\theta} \prod_{n=1}^{N} p_{n,\theta}(\boldsymbol{x}) \iff \log p_\theta(\boldsymbol{x}) = \underbrace{\sum_{n=1}^{N} \log p_{n,\theta}(\boldsymbol{x})}_{-E_\theta(\boldsymbol{x})} - \log Z_\theta. \quad (4)$$

We want to train the model via maximum-likelihood

$$\theta^* = \arg\max_\theta \log p_\theta(\boldsymbol{x}), \quad (5)$$

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

$$\nabla_\theta \log p_\theta(\boldsymbol{x}) = -E_\theta(\boldsymbol{x}) - \nabla_\theta \log Z_\theta \quad (6)$$

$$\nabla_\theta \log Z_\theta = -\nabla_\theta \mathbb{E}_{\boldsymbol{x} \sim p_\theta(\boldsymbol{x})} [\nabla_\theta E_\theta(\boldsymbol{x})] \quad (7)$$

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

Training PoEs by Constrastive Divergence

Contrastive divergence is a cancellation trick plus an approximation.

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

$$\theta^* = \arg\max_\theta \log p_\theta(\boldsymbol{x}) \iff \theta^* = \arg\min_\theta KL(p^0||p_\theta) \tag{8}$$

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

$$\boldsymbol{x} \sim p_\theta^t(\boldsymbol{x}) \iff \boldsymbol{x} \sim \texttt{MCMC}(\text{target} = p_\theta, \text{init} = \boldsymbol{x}_0), \boldsymbol{x}_0 \sim p^0(\boldsymbol{x}). \tag{9}$$

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

$$\Delta KL = KL(p^0||p_\theta^\infty) - KL(p_\theta^t||p_\theta^\infty) \tag{10}$$

Motivation: Pesky term from Z_θ cancels.

Training PoEs by Constrastive Divergence

$$\Delta KL = \underbrace{KL(p^0||p_\theta^\infty)}_{①} - \underbrace{KL(p_\theta^t||p_\theta^\infty)}_{②}. \tag{11}$$

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

$$KL(p_\theta^t||p_\theta^\infty) \leq KL(p^0||p_\theta^\infty) \implies \Delta KL \geq 0 \tag{12}$$

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

$$p_\theta^\infty = p^0 \implies p_\theta^t = p^0, \Delta KL = 0 \tag{13}$$

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

$$\Delta KL \to KL(p^0||p_\theta^\infty), \text{ as } t \to \infty. \tag{14}$$

Intuition: ΔKL encourages ① p_θ^∞ close to p^0 and ② p_θ^t far from p^0. Since p_θ^t is starts at p^0, ② encourages the chain to not wander from p^0.

Training PoEs by Contrastive Divergence

Cancellation: Let's see how the Z_θ term cancels

$$\nabla_\theta \, \Delta KL = \nabla_\theta \left[KL(p^0 || p_\theta^\infty) - KL(p_\theta^t || p_\theta^\infty) \right] \tag{15}$$

$$= \int \left[\cancel{\frac{dp^0}{d\theta} \frac{\delta}{\delta p^0} KL(p^0 || p_\theta^\infty)} + \frac{dp_\theta^\infty}{d\theta} \frac{\delta}{\delta p_\theta^\infty} KL(p^0 || p_\theta^\infty) + \right.$$

$$\left. - \frac{dp_\theta^t}{d\theta} \frac{\delta}{\delta p_\theta^t} KL(p_\theta^t || p_\theta^\infty) - \frac{dp_\theta^\infty}{d\theta} \frac{\delta}{\delta p_\theta^\infty} KL(p_\theta^t || p_\theta^\infty) \right] d\boldsymbol{x} \tag{16}$$

$$\frac{dp_\theta^\infty}{d\theta} \frac{\delta}{\delta p_\theta^\infty} KL(p^0 || p_\theta^\infty) = -p^0 \frac{d \log p_\theta^\infty}{d\theta} \tag{17}$$

$$\frac{dp_\theta^\infty}{d\theta} \frac{\delta}{\delta p_\theta^\infty} KL(p_\theta^t || p_\theta^\infty) = -p_\theta^t \frac{d \log p_\theta^\infty}{d\theta} \tag{18}$$

Now we also have that

$$\frac{d \log p_\theta^\infty}{d\theta} = -\frac{dE_\theta}{d\theta} - \frac{d \log Z_\theta}{d\theta}. \tag{19}$$

The terms involving Z_θ cancel!

Training PoEs by Contrastive Divergence

The cancellation leaves us with

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

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

The second term can be estimated with simple Monte Carlo.

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

$$\nabla_\theta \, \Delta KL \approx \mathbb{E}_{\boldsymbol{x} \sim p^0(\boldsymbol{x})} \left[\frac{dE_\theta(\boldsymbol{x})}{d\theta} \right] - \mathbb{E}_{\boldsymbol{x} \sim p_\theta^t(\boldsymbol{x})} \left[\frac{dE_\theta(\boldsymbol{x})}{d\theta} \right]. \tag{20}$$

Empirically shows that this update also reduces the ignored term.

(Restricted) Boltzmann Machines

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

$$E_\theta(\boldsymbol{x}, \boldsymbol{h}) = \sum_{i \in X, j \in H} x_i h_j w_{ij} + \sum_{i \in X} x_i \theta_i + \sum_{i \in H} h_i \theta_i + \tag{21}$$
$$+ \sum_{i<j \in X} x_i x_j w_{ij} + \sum_{i<j \in H} h_i h_j w_{ij}.$$

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

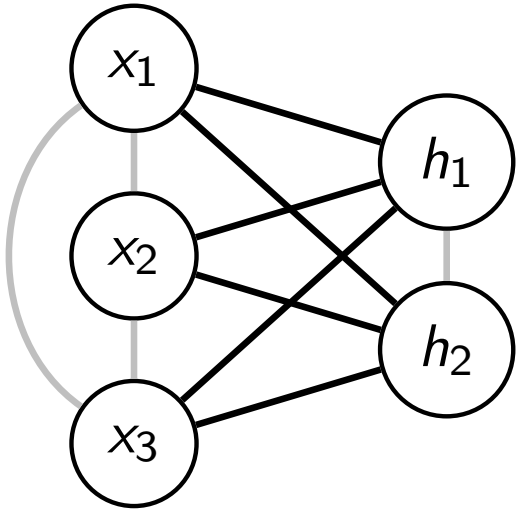


Figure 2: Graphical model of a (restricted) Boltzmann Machine.

(Restricted) Boltzmann Machines

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

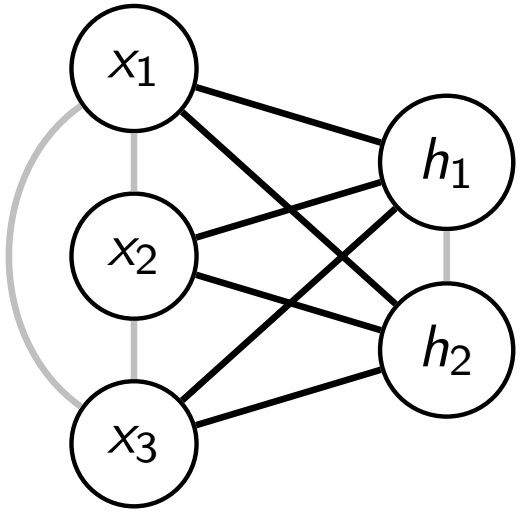


Figure 3: Graphical model of a (restricted) Boltzmann Machine.

Freund and Haussler [1994]: RBMs are PoEs $\implies$ still intractable.

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

(Restricted) Boltzmann Machines

$$\nabla_\theta \, \Delta KL \approx \mathbb{E}_{\boldsymbol{x} \sim p^0(\boldsymbol{x})} \left[\frac{dE_\theta(\boldsymbol{x})}{d\theta} \right] - \mathbb{E}_{\boldsymbol{x} \sim p_\theta^t(\boldsymbol{x})} \left[\frac{dE_\theta(\boldsymbol{x})}{d\theta} \right]$$

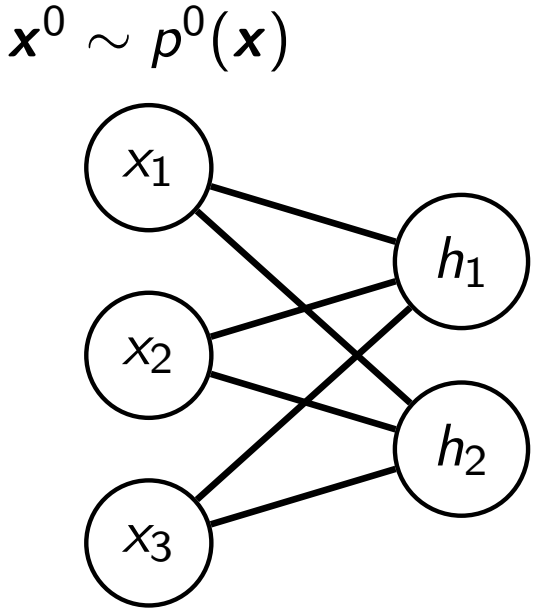


Figure 4: Gibbs sampling in an RBM for contrastive divergence.

(Restricted) Boltzmann Machines

$$\nabla_\theta \, \Delta KL \approx \mathbb{E}_{\boldsymbol{x} \sim p^0(\boldsymbol{x})} \left[\frac{dE_\theta(\boldsymbol{x})}{d\theta} \right] - \mathbb{E}_{\boldsymbol{x} \sim p_\theta^t(\boldsymbol{x})} \left[\frac{dE_\theta(\boldsymbol{x})}{d\theta} \right]$$

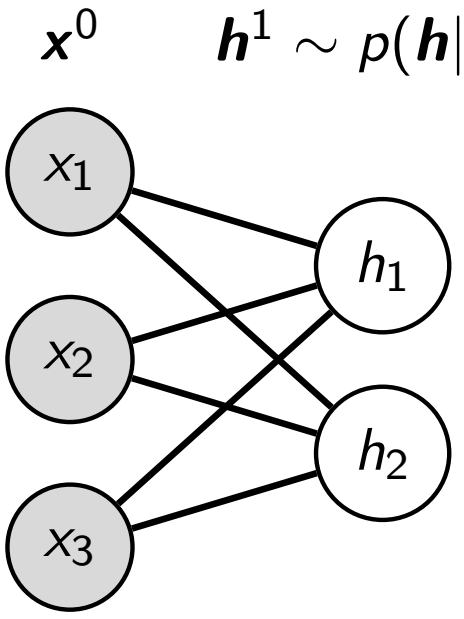


Figure 5: Gibbs sampling in an RBM for contrastive divergence.

RBMs (generally PoEs): easy to sample $p(\boldsymbol{x}|\boldsymbol{h})$ and $p(\boldsymbol{h}|\boldsymbol{x})$.

(Restricted) Boltzmann Machines

$$\nabla_\theta \, \Delta KL \approx \mathbb{E}_{\boldsymbol{x}\sim p^0(\boldsymbol{x})}\left[\frac{dE_\theta(\boldsymbol{x})}{d\theta}\right] - \mathbb{E}_{\boldsymbol{x}\sim p^t_\theta(\boldsymbol{x})}\left[\frac{dE_\theta(\boldsymbol{x})}{d\theta}\right]$$

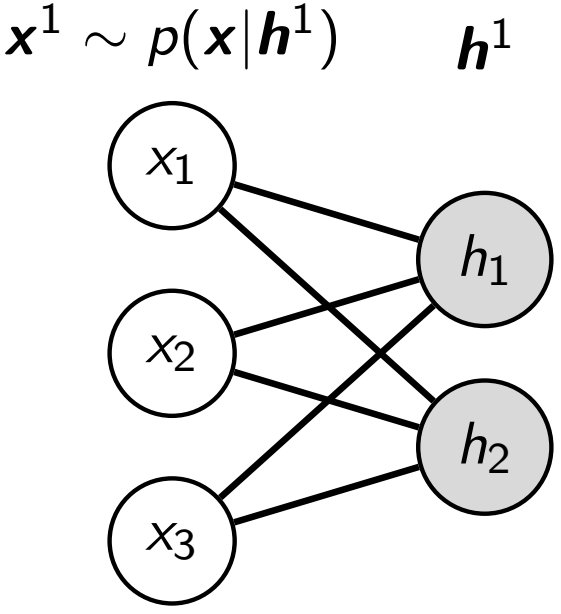


Figure 6: Gibbs sampling in an RBM for contrastive divergence.

RBM (generally PoEs): easy to sample $p(\boldsymbol{x}|\boldsymbol{h})$ and $p(\boldsymbol{h}|\boldsymbol{x})$.

Some results

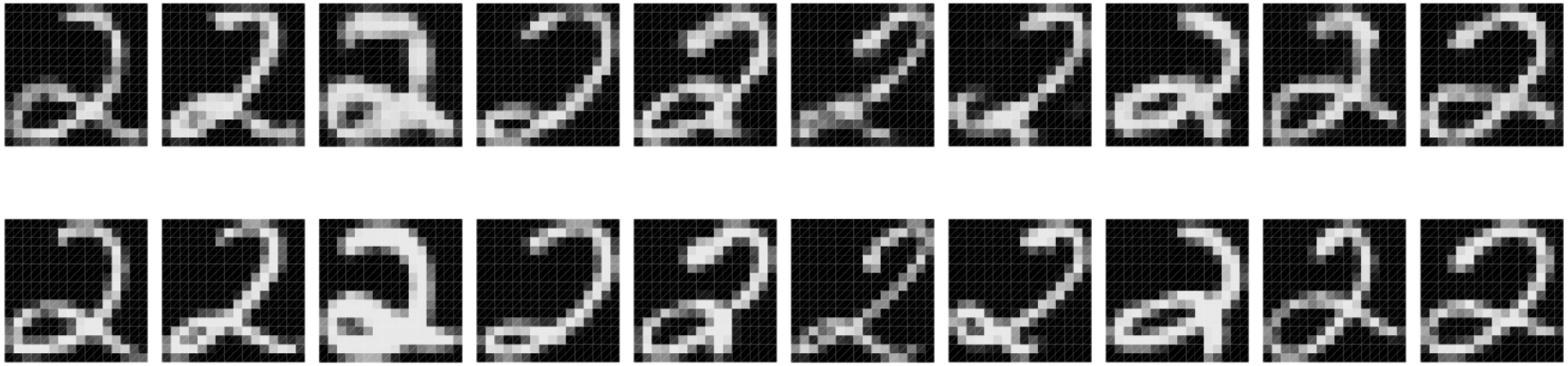


Figure 7: MNIST images (top) and their reconstrutions (bottom) by an RBM.

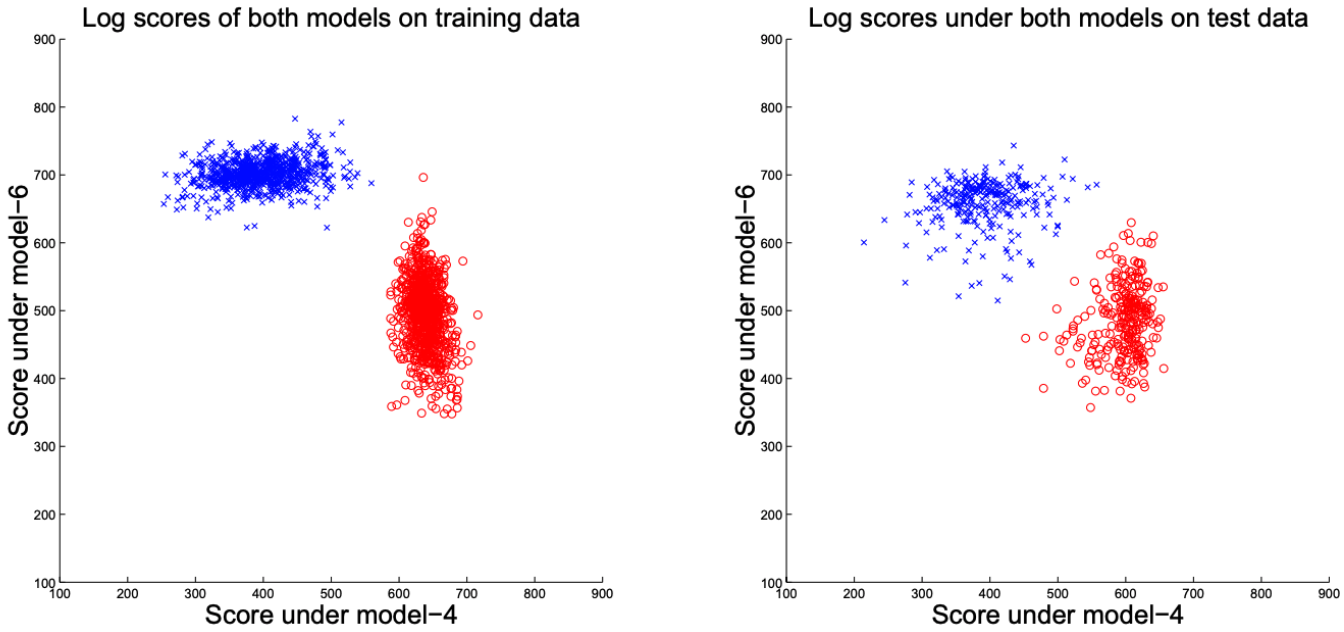


Figure 8: Energies of digits 4 and 6 under RBMs trained with digits 4 and 6.

Deep Boltzmann Machines

Deep BMs: Stack multiple RBMs [Salakhutdinov and Hinton, 2009].

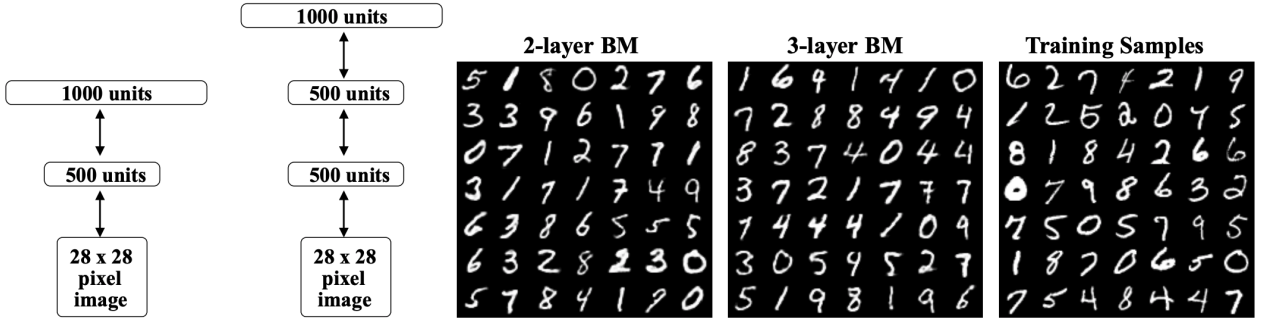


Figure 9: Training and generated data using a DBM on MNIST.

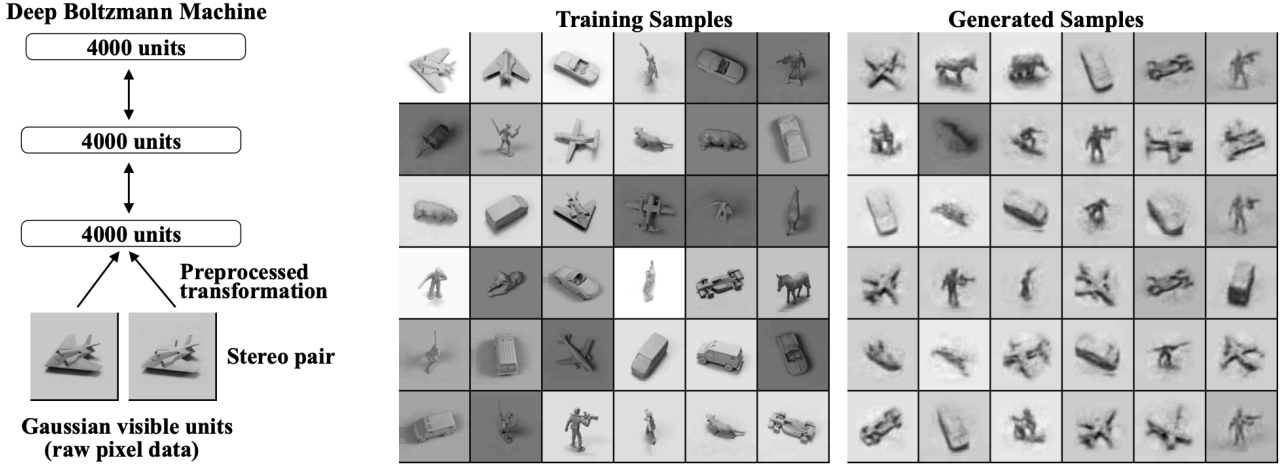


Figure 10: Training and generated data using a DBM on NORB.

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

Modern Training – Contrastive Divergence I

We want to maximise the likelihood

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{\exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))}{Z_{\boldsymbol{\theta}}} \tag{22}$$

but we can't compute the normalizing constant

$$Z_{\boldsymbol{\theta}} = \int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x})) \, \mathrm{d}\mathbf{x}. \tag{23}$$

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

$$\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) = -\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}}. \tag{24}$$

Modern Training – Contrastive Divergence II

The first term $\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})$ is easy to compute with AD. But, the second term $\nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}}$ is intractable to compute exactly. However, it can be approximated, with simple MC, as

$$\nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}} = \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x})} \left[-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}) \right] \approx \frac{1}{N} \sum_{n}^{N} -\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}_n), \ \mathbf{x}_n \sim p_{\boldsymbol{\theta}}(\mathbf{x}). \tag{25}$$

Thus, we are taking gradient steps in the direction

$$\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}_{\text{train}}) - \nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}_{\text{sample}}). \tag{26}$$

Modern Training – Contrastive Divergence III

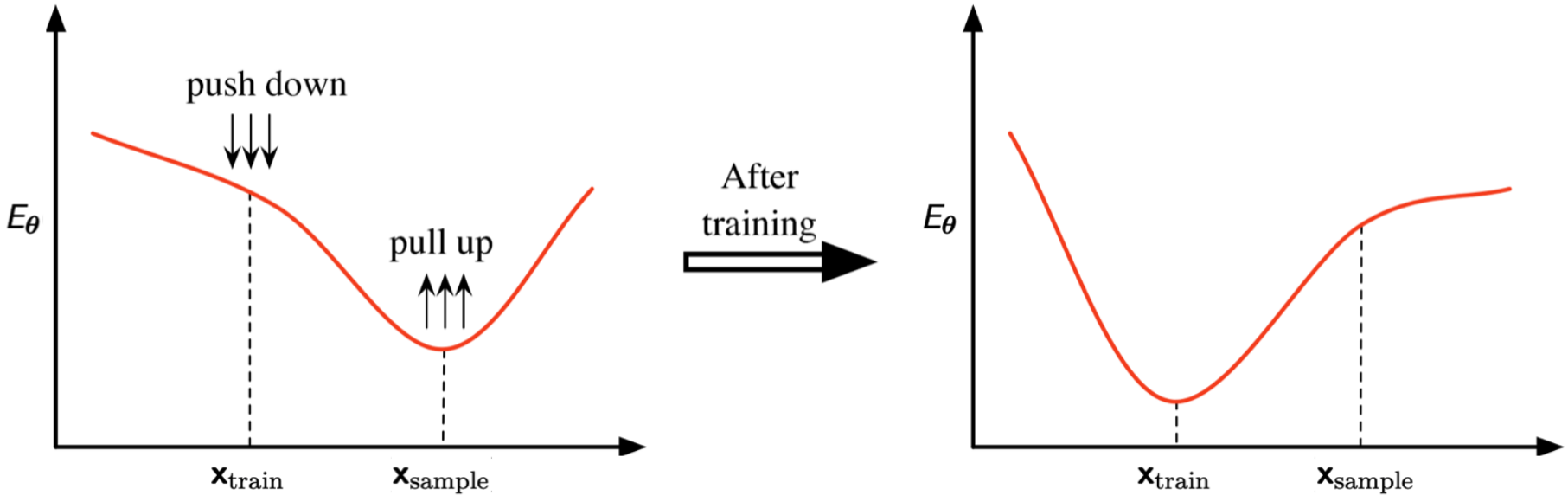


Figure 11: Taking steps in the direction of $\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}_{\text{train}}) - \nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}_{\text{sample}})$. Adapted from [LeCun et al., 2006].

Modern Training – Contrastive Divergence IV

Alas, sampling from $p_{\boldsymbol{\theta}}(\mathbf{x})$ is highly *non-trivial*. Thus, we must resort to further approximation. A common choice is to use Langevin MCMC

$$\mathbf{x}^{t+1} \leftarrow \mathbf{x}^t + \frac{\epsilon^2}{2}\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}^t) + \epsilon \mathbf{z}^t, \quad t = 0, 1, \cdots, T-1, \tag{27}$$

where $\mathbf{z}^t \sim \mathcal{N}(0, 1)$ and $\mathbf{x}^0 \sim p(\mathbf{x})$.
Note: $\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}^t) = -\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}}(\mathbf{x}) - \cancel{\nabla_{\mathbf{x}} \log Z_{\boldsymbol{\theta}}} = -\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}}(\mathbf{x})$.

When $\epsilon \to 0$ and $T \to \infty$, $\mathbf{x}^T \sim p_{\boldsymbol{\theta}}(\mathbf{x})$.
However, we usually do not run the chain to convergence due to the high computational cost.

Improved CD for EBMs I

But wait! The CD gradient doesn't come from the LL, but rather

$$\Delta KL = KL(p^0||p_\theta^\infty) - KL(p_\theta^t||p_\theta^\infty). \tag{28}$$

Taking gradients w.r.t this objective results in

$$-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}) + \mathbb{E}_{\mathbf{x}\sim p_\theta^t(\mathbf{x})}\left[\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})\right] - \nabla_{\boldsymbol{\theta}} p_\theta^t(\mathbf{x}) \nabla_{p_\theta^t(\mathbf{x})} KL(p_\theta^t||p_\theta^0) \tag{29}$$

where we have an additional KL term [Du et al., 2020].

CD training without the KL term doesn't minimize any scalar loss function [Sutskever and Tieleman, 2010].

Improved CD for EBMs II

Du et al. [2020] show that Adding the KL term is equivalent to adding a KL loss

$$\mathcal{L}_{KL} = \underbrace{\mathbb{E}_{p_\theta^t(\mathbf{x})}[\mathbf{x}]}_{①} + \underbrace{\mathbb{E}_{p_\theta^t(\mathbf{x})}\left[\log p_\theta^t(\mathbf{x})\right]}_{②}. \tag{30}$$

Estimation of the KL loss is fairly involved.

① requires differentiating through the MCMC sampling.

② is estimated via a nearest-neighbours approximation.

Improved CD for EBMs III

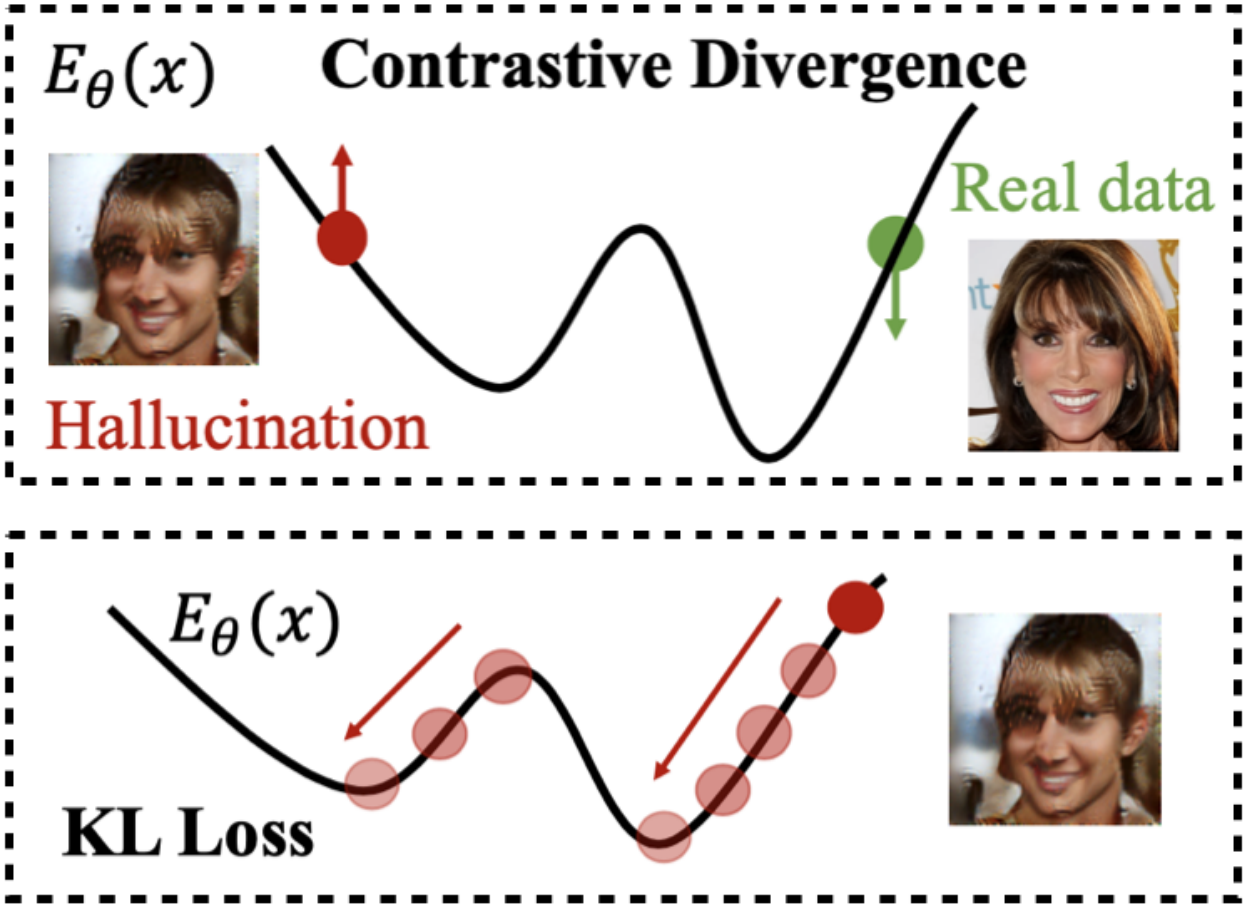


Figure 12: Intuition for the KL loss – regularization which prevents bad samples! Adapted from [Du et al., 2020].

Improved CD for EBMs IV

Figure 13: Samples from an EBM composed of individual conditional EBMs trained on the CelebA-HQ dataset using Improved CD. Adapted from [Du et al., 2020].

Modern Training – Score Matching

If $f(\mathbf{x})$ and $g(\mathbf{x})$ have equal first derivatives (a.k.a *score functions*), then $f(\mathbf{x}) \equiv g(\mathbf{x}) + \text{constant}$. When $f(\mathbf{x})$ and $g(\mathbf{x})$ are log PDFs, $f(\mathbf{x}) \equiv g(\mathbf{x})$. Thus, Hyvärinen and Dayan [2005] propose to learn an EBM by minimising

$$D_F(p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\frac{1}{2}\left\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})\right\|^2\right]. \tag{31}$$

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

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

$$D_F(p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\frac{1}{2}\sum_{i=1}^{d}\left(\frac{\partial E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i}\right)^2 + \frac{\partial^2 E_{\boldsymbol{\theta}}(\mathbf{x})}{(\partial x_i)^2}\right] + \text{C}. \tag{32}$$

Modern Training – Denoising Score Matching

Naive Score Matching has two potentially 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 $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\epsilon}$, 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:

$$\begin{aligned} D_F(q(\tilde{\mathbf{x}}) \parallel p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})) &= \mathbb{E}_{q(\tilde{\mathbf{x}})} \left[\frac{1}{2} \left\| \nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}}) - \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) \right\|_2^2 \right] \\ &= \mathbb{E}_{q(\mathbf{x}, \tilde{\mathbf{x}})} \left[\frac{1}{2} \left\| \nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}} | \mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) \right\|_2^2 \right] + \mathrm{C}, \end{aligned} \tag{33}$$

thus avoiding any expensive second-order gradients.

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

Modern Training – Sliced Score Matching

Sliced Score Matching minimizes the sliced Fisher divergence

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

where $p(\mathbf{v})$ denotes a projection dist. such that $\mathbb{E}_{p(\mathbf{v})}[\mathbf{v}\mathbf{v}^\mathsf{T}]$ is pos. def.

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

$$\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{p(\mathbf{v})}\left[\frac{1}{2}\sum_{i=1}^{d}\left(\frac{\partial E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i}v_i\right)^2 + \sum_{i=1}^{d}\sum_{j=1}^{d}\frac{\partial^2 E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i \partial x_j}v_i v_j\right]. \tag{34}$$

However, unlike before, this form has a computational complexity of $\mathcal{O}(d)$ rather than $\mathcal{O}(d^2)$:

$$\sum_{i=1}^{d}\sum_{j=1}^{d}\frac{\partial^2 E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i \partial x_j}v_i v_j = \sum_{i=1}^{d}\frac{\partial}{\partial x_i}\underbrace{\left(\sum_{j=1}^{d}\frac{\partial E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_j}v_j\right)}_{:=f(\mathbf{x})}v_i. \tag{35}$$

Modern Training – Noise Contrastive Estimation

Gutmann and Hyvärinen [2010] proposed an alternative training method.

Define a known and tractable reference distribution $p_r(\boldsymbol{x})$.

Treat Z_θ as a trainable variable.

$$\log p_\theta = \log \tilde{p}_\theta(\boldsymbol{x}) + C. \tag{36}$$

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

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

$$p(y = 0|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{p_r(\boldsymbol{x})}{p_\theta(\boldsymbol{x}) + p_r(\boldsymbol{x})}, \; p(y = 1|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{x}) + p_r(\boldsymbol{x})}, \tag{37}$$

where $y = 0, 1$ mean $\boldsymbol{x}$ drawn from $p_r(\boldsymbol{x})$ and from $p_D(\boldsymbol{x})$ respectively.

Modern Training – Noise Contrastive Estimation

Draw $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \sim p_r(\boldsymbol{x})$ and $\boldsymbol{x}_{N+1}, \ldots, \boldsymbol{x}_{2N} \sim p_D(\boldsymbol{x})$. Minimise

$$\mathcal{L}_{NCE} = \sum_{n=1}^{N} \log p(y_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta}) + \log p(y_{N+n} = 0|\boldsymbol{x}_{N+n}, \boldsymbol{\theta}), \tag{38}$$

binary cross entropy loss for classifying samples.

Theorem (informal) Gutmann and Hyvärinen [2010]

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

$$p(y = 0|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{p_r(\boldsymbol{x})}{p_\theta(\boldsymbol{x}) + p_r(\boldsymbol{x})}, p(y = 1|\boldsymbol{x}, \ \boldsymbol{\theta}) = \frac{p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{x}) + p_r(\boldsymbol{x})}, \tag{39}$$

Intuition: Trainable variable C cannot go to neither $-\infty$ nor ∞.

Conditional Noise Contrastive Estimation

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

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

$$D(\boldsymbol{x}, \boldsymbol{x}') = p(y = 1|\boldsymbol{x}, \boldsymbol{x}', \theta) = \frac{q(\boldsymbol{x}'|\boldsymbol{x})p_\theta(\boldsymbol{x})}{q(\boldsymbol{x}'|\boldsymbol{x})p_\theta(\boldsymbol{x}) + q(\boldsymbol{x}|\boldsymbol{x}')p_\theta(\boldsymbol{x}')}, \tag{40}$$

where a typical choice for q is

$$q(\boldsymbol{x}'|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}'; \boldsymbol{x}, \sigma^2\boldsymbol{I}). \tag{41}$$

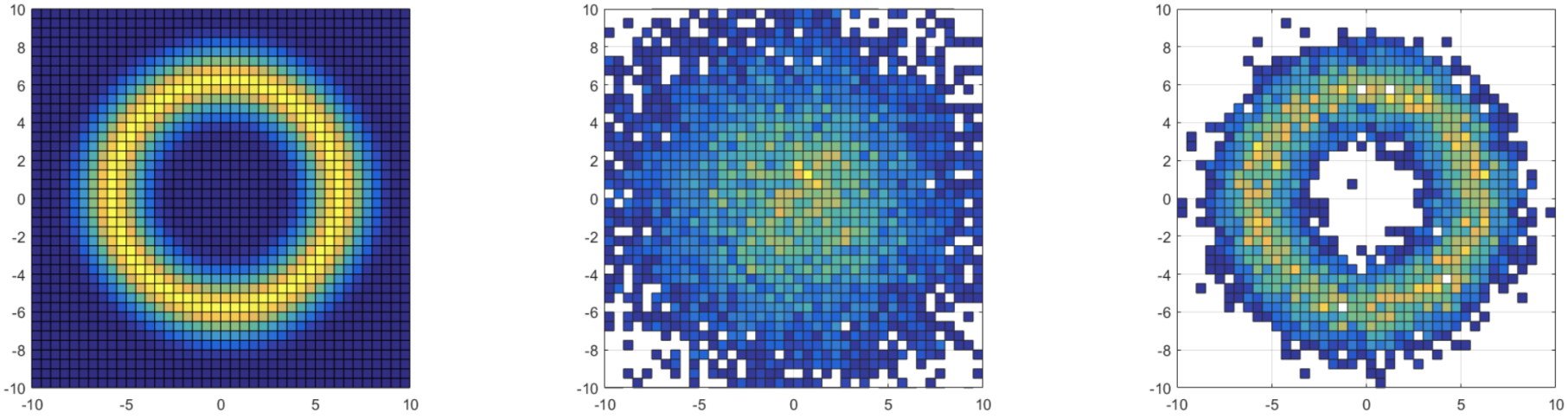
Observation: Equation (40) invariant to scaling p_θ, and Z_θ cancels.

Similarly to NCE, set up classification task and minimise

$$\mathcal{L}_{CNCE} = -\sum_{n=1}^{N} \left[\log D(\boldsymbol{x}_n, \boldsymbol{x}'_n) + \log(1 - D(\boldsymbol{x}'_n, \boldsymbol{x}_n))\right] \tag{42}$$

where $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N \sim p_D(\boldsymbol{x})$ and $\boldsymbol{x}'_1, \cdots, \boldsymbol{x}'_N \sim q(\boldsymbol{x}'|\boldsymbol{x})$.

Conditional Noise Contrastive Estimation



(a) Contour plot of the data pdf

(b) NCE noise (histogram)

(c) CNCE noise (histogram)

Figure 14: Data distribution, NCE samples and CNCE samples.

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 [2020] view CD as a (C)NCE classification problem.

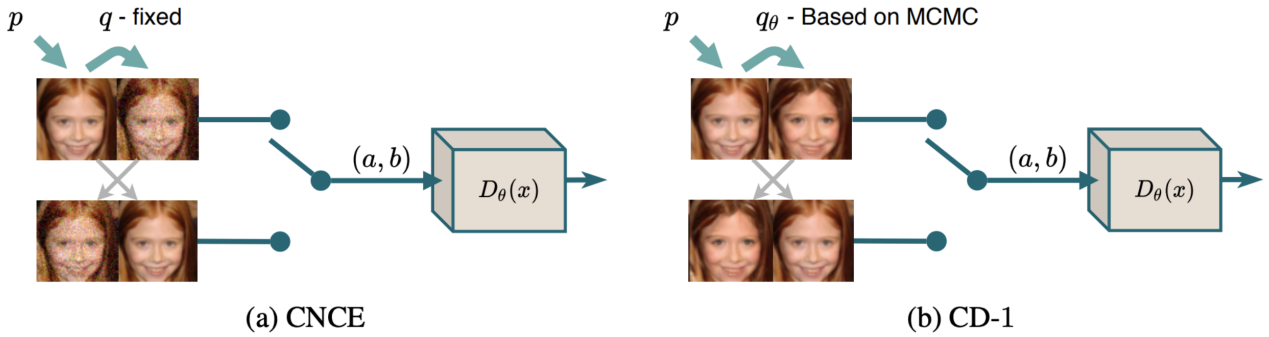


Figure 15: Illustration of CD viewed as a classification problem.

The CNCE parameter update rule can be written as

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

Intuition:

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

Relation between (C)NCE and CD

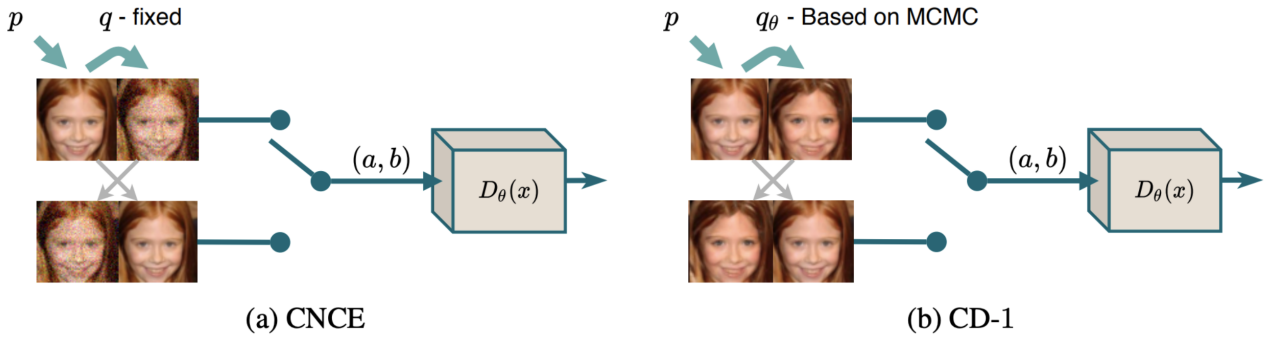


Figure 16: Illustration of CD viewed as a classification problem.

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

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

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

The optimal classifier becomes a random guess

$$D(\boldsymbol{x}_n, \boldsymbol{x}'_n) = \frac{q(\boldsymbol{x}'|\boldsymbol{x})p_\theta(\boldsymbol{x})}{q(\boldsymbol{x}'|\boldsymbol{x})p_\theta(\boldsymbol{x}) + q(\boldsymbol{x}|\boldsymbol{x}')p_\theta(\boldsymbol{x}')} = \frac{1}{2}. \tag{43}$$

Relation between (C)NCE and CD

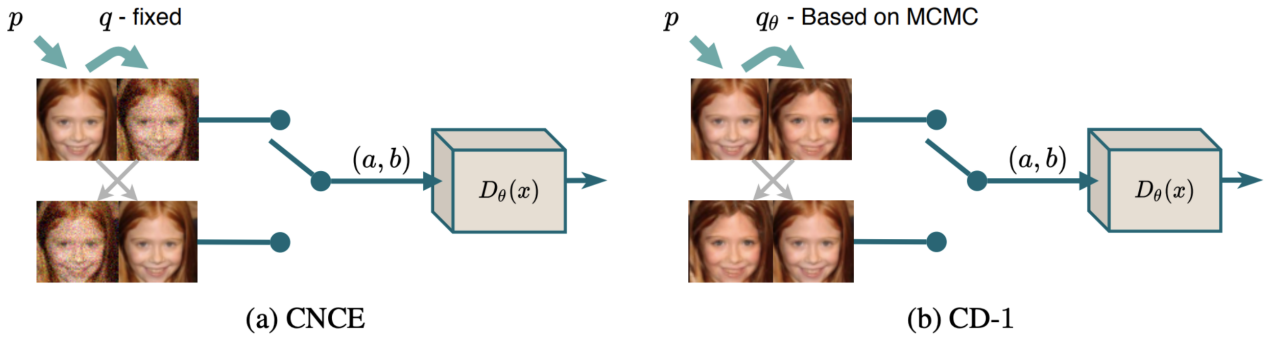


Figure 17: Illustration of CD viewed as a classification problem.

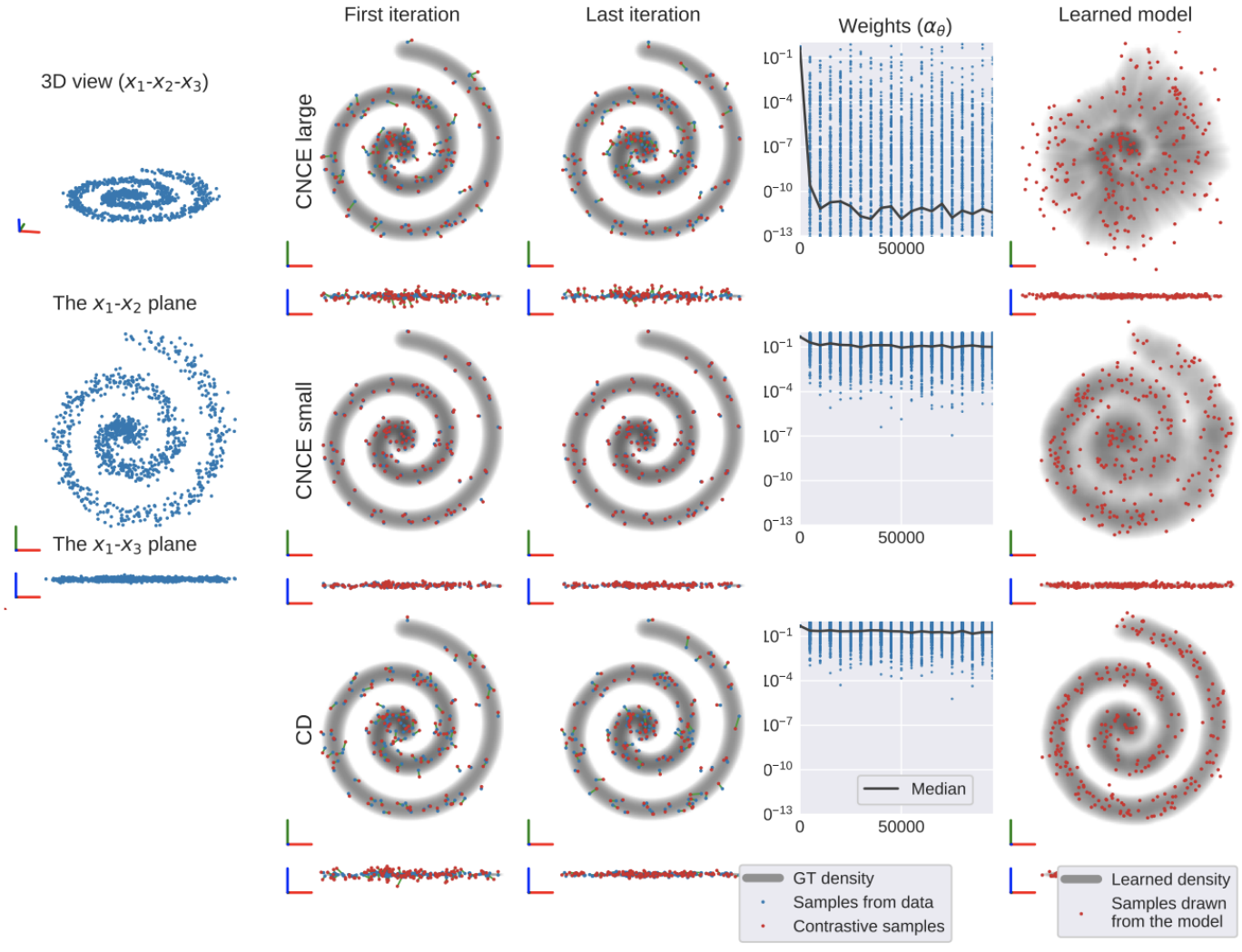
Under this model, the update rule becomes identical to CD

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

Important detail (stopping gradients)

Sample $\boldsymbol{x}'_n \sim q_\theta(\boldsymbol{x}'|\boldsymbol{x})$ 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.

Relation between (C)NCE and CD



(a) The toy model

(b) Comparing CNCE with CD

Figure 18: Illustration of CNCE and CD training.

Relation between SM and CD

Hyvarinen [2007] relates SM with CD under a particular Markov Chain.

Suppose we generate samples using Langevin Dynamics

$$\boldsymbol{x}' = \boldsymbol{x} + \frac{\eta^2}{2}\nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}) + \eta\epsilon, \tag{44}$$

where $\eta > 0$ is a step size and ϵ is standard Gaussian noise. Starting from the CD update rule, if we

1. Stop gradients flowing though p_θ in equation (44),
2. Take the limit $\eta \to 0$,

we recover the SM update rule.

Theorem (informal) Hyvarinen [2007]

Under an appropriate gradient stopping approximation, the update rule of CD is equivalent to the update rule of SM in the limit $\eta \to 0$.

EBMs in the Wild

Classifiers are EBMs? [Grathwohl et al., 2020b] I

Classifier:

$$p_\theta(y \mid \mathbf{x}) = \frac{\exp\left(f_\theta(\mathbf{x})[y]\right)}{\sum_{y'} \exp\left(f_\theta(\mathbf{x})[y']\right)}. \quad (45)$$

Joint:

$$p_\theta(\mathbf{x}, y) = \frac{\exp\left(f_\theta(\mathbf{x})[y]\right)}{Z_\theta}. \quad (46)$$

Marginal:

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y) = \frac{\sum_y \exp\left(f_\theta(\mathbf{x})[y]\right)}{Z_\theta}.$$

Energy:

$$E_\theta(\mathbf{x}) = -\log \sum_y \exp(f_\theta(\mathbf{x})[y]). \quad (47)$$

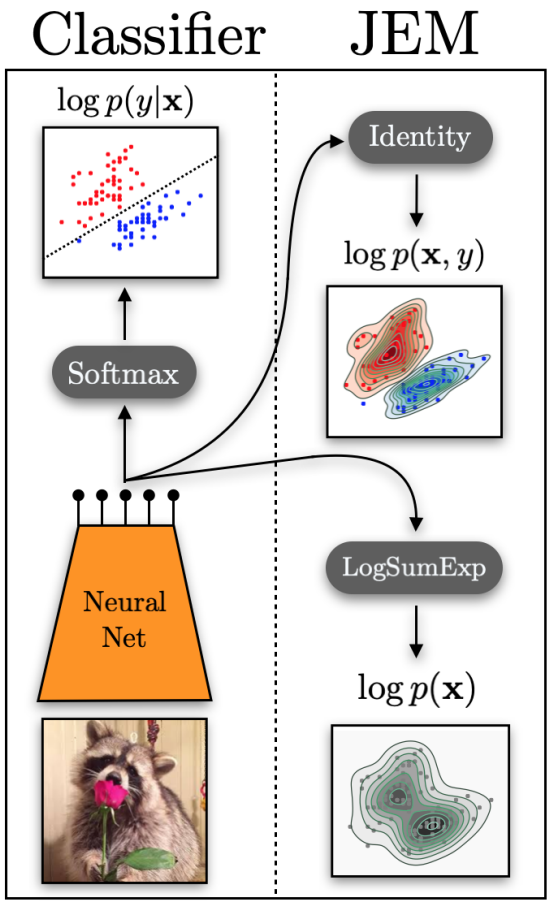


Figure 19: JEM.

Classifiers are EBMs? [Grathwohl et al., 2020b] II

Note:

$$p_\theta(y \mid \mathbf{x}) = \frac{p_\theta(\mathbf{x}, y)}{p_\theta(\mathbf{x})} = \frac{\exp\left(f_\theta(\mathbf{x})[y]\right)}{\cancel{Z_\theta}} \frac{\cancel{Z_\theta}}{\sum_{y}' \exp\left(f_\theta(\mathbf{x})[y']\right)}, \tag{48}$$

as before.
For optimization, factorise the joint:

$$\log p_\theta(\mathbf{x}, y) = \log p_\theta(\mathbf{x}) + \log p_\theta(y|\mathbf{x}). \tag{49}$$

$\log p_\theta(\mathbf{x})$ is optimised using persistent CD.

$\log p_\theta(y|\mathbf{x})$ is optimized using the standard CE loss.

JEM is a good classifier *and* generative model. The classifier is well calibrated, and the generative model knows what it doesn't know.

VAE-EBM Hybrids [Xiao et al., 2020] I

Product of a VAE and an EBM: $h_{\psi,\theta}(\mathbf{x},\mathbf{z}) = \frac{1}{Z_{\psi,\theta}} p_\theta(\mathbf{x},\mathbf{z}) e^{-E_\psi(\mathbf{x})}$.

Marginalizing out $\mathbf{z}$ gives

$$h_{\psi,\theta}(\mathbf{x}) = \frac{1}{Z_{\psi,\theta}} \int p_\theta(\mathbf{x},\mathbf{z}) e^{-E_\psi(\mathbf{x})} d\mathbf{z} = \frac{1}{Z_{\psi,\theta}} p_\theta(\mathbf{x}) e^{-E_\psi(\mathbf{x})}. \tag{50}$$

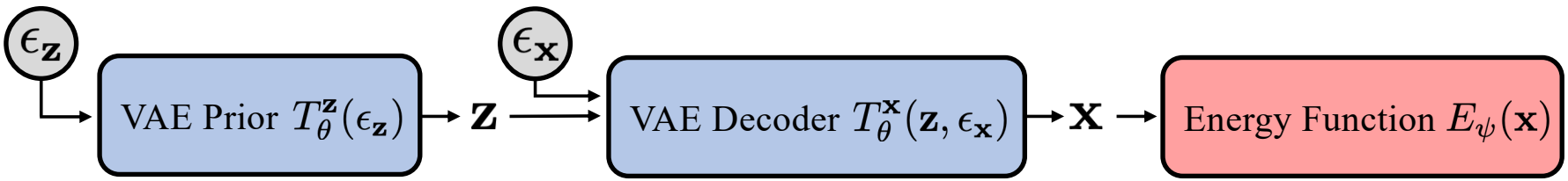


Figure 20: VAEBM Computational Model

VAE-EBM Hybrids [Xiao et al., 2020] II

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

$$\begin{aligned}\log h_{\psi,\theta}(\mathbf{x}) &= \log p_\theta(\mathbf{x}) - E_\psi(\mathbf{x}) - \log Z_{\psi,\theta} \qquad (51)\\ &\geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))}_{\mathcal{L}_{\text{vae}}(\mathbf{x},\theta,\phi)} \underbrace{-E_\psi(\mathbf{x}) - \log Z_{\psi,\theta}}_{\mathcal{L}_{\text{EBM}}(\mathbf{x},\psi,\theta)}.\end{aligned}$$

Step 1: Train the VAE via $\mathcal{L}_{\text{vae}}$.

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

Figure 21: CelebA HQ 256 – Qualitative Results

Energy-based OOD Detection [Liu et al., 2020]

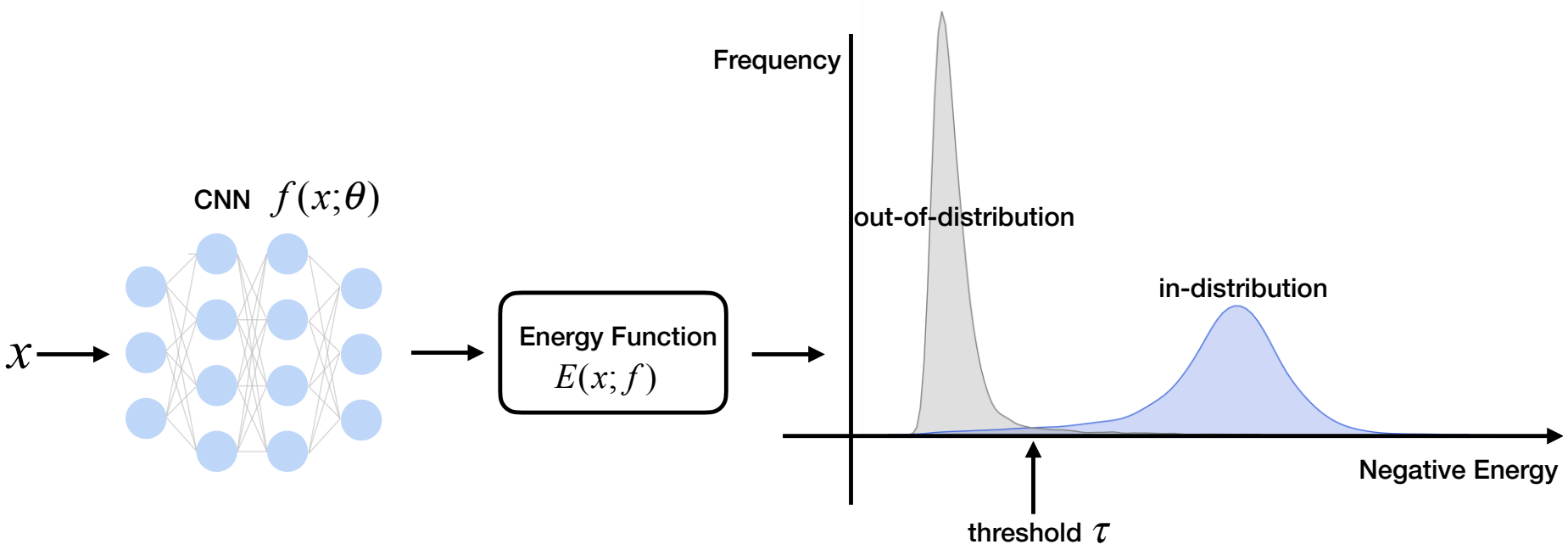


Figure 22: Energy-based OOD Detection.

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

Outperforms JEM for OOD detection.

Conclusions

EBMs are flexible class of models, which expand our modelling toolbox.

Several approaches for training EBMs, each with pros and cons:

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].)

Conclusions

Energy Based Models are great!

Many things that we didn't get to talk about...

Other training methods:

- Minimizing KL differences – generalized CD and SM.
- Minimizing Stein's discrepancy. E.g. LSD [Grathwohl et al., 2020c].
- Adversarial training

Other cool EBM papers:

- Your GAN is also a secret EBM [Che et al., 2020].
- Generalised Energy Based Models (GAN-EBM hybrids) [Arbel et al., 2020].

And of course, many more details on CD, DM, NCE, their variants, their connections, and more in [Song and Kingma, 2021].

References I

M. Arbel, L. Zhou, and A. Gretton. Generalized energy based models. *arXiv preprint arXiv:2003.05033*, 2020.

C. Ceylan and M. U. Gutmann. Conditional noise-contrastive estimation of unnormalised models. In *International Conference on Machine Learning*, pages 726–734. PMLR, 2018.

T. Che, R. Zhang, J. Sohl-Dickstein, H. Larochelle, L. Paull, Y. Cao, and Y. Bengio. Your gan is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*, 2020.

Y. Du, S. Li, J. Tenenbaum, and I. Mordatch. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020.

Y. Freund and D. Haussler. Unsupervised learning of distributions of binary vectors using two layer networks. 1994.

References II

W. Grathwohl, J. Kelly, M. Hashemi, M. Norouzi, K. Swersky, and D. Duvenaud. No mcmc for me: Amortized sampling for fast and stable training of energy-based models. *arXiv preprint arXiv:2010.04230*, 2020a.

W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky. Your classifier is secretly an energy based model and you should treat it like one, 2020b.

W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, and R. Zemel. Learning the stein discrepancy for training and evaluating energy-based models without sampling. In *International Conference on Machine Learning*, pages 3732–3747. PMLR, 2020c.

References III

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

G. E. Hinton, T. J. Sejnowski, et al. Learning and relearning in boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(282-317):2, 1986.

A. Hyvarinen. Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on neural networks*, 18(5):1529–1531, 2007.

References IV

A. Hyvärinen and P. Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6 (4), 2005.

Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

W. Liu, X. Wang, J. D. Owens, and Y. Li. Energy-based out-of-distribution detection. *arXiv preprint arXiv:2010.03759*, 2020.

R. Salakhutdinov and G. Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455. PMLR, 2009.

P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, Colorado Univ at Boulder Dept of Computer Science, 1986.

Y. Song and D. P. Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

References V

I. Sutskever and T. Tieleman. On the convergence properties of contrastive divergence. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 789–795. JMLR Workshop and Conference Proceedings, 2010.

P. Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Z. Xiao, K. Kreis, J. Kautz, and A. Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models. In *International Conference on Learning Representations*, 2020.

O. Yair and T. Michaeli. Contrastive divergence learning is a time reversal adversarial game. *arXiv preprint arXiv:2012.03295*, 2020.