Notes on Training of Energy-Based Models

Jiachun Jin

School of Information Science and Technology

ShanghaiTech University

Energy-based models (EBMs), also known as unnormalized models, are quite flexible for probabilistic modeling. In this note I mainly supplementing the skipped derivation details of [Song and Kingma, 2021].

Contents

1 Energy-based models

The density given by an EBM is:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{\exp\left(-E_{\boldsymbol{\theta}}(\mathbf{x})\right)}{Z(\boldsymbol{\theta})}, \tag{1}$$

where $E_{\boldsymbol{\theta}}(\mathbf{x})$ is called the energy, and $Z_{\boldsymbol{\theta}} = \int \exp\left(-E_{\boldsymbol{\theta}}(\mathbf{x})\right)$ denotes the normalizing constant, which is intractable.

2 Training with MLE

Maximum likelihood estimation (MLE) is the *de facto* standard for learning probabilistic models from i.i.d. data. The gradient of the log likelihood with respect to the model parameter $\boldsymbol{\theta}$ is given by:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) &= -\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}} \\
&= -\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\boldsymbol{\theta}} \log \int \exp\{-E_{\boldsymbol{\theta}}(\mathbf{x})\} \mathrm{d}\mathbf{x} \\
&= -\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}) - \frac{1}{Z_{\boldsymbol{\theta}}} \int \nabla_{\boldsymbol{\theta}} \exp\{-E_{\boldsymbol{\theta}}(\mathbf{x})\} \mathrm{d}\mathbf{x} \\
&= -\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}) + \int p_{\boldsymbol{\theta}}(\mathbf{x}) \nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} \\
&= \underbrace{-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})}_{\text{positive phase}} + \underbrace{\mathbb{E}_{\tilde{\mathbf{x}} \sim p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})} [\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})]}_{\text{negative phase}},
\end{aligned} \tag{2}$$

in the third line, we switch the order of gradient and integral, the details about the validity is discussed in section 2.4 of [Casella and Berger, 2002]. The positive phase tries to decrease the energy of real data samples, and the negative phase tries to increase the energy of sample generated by the current model (this can be considered as reducing the model's incorrect beliefs about the world, which is analogous to what human beings do when they are dreaming) [Goodfellow et al., 2016, Section 18.2]. The "wake-sleep" fashion can be used in approximate inference settings [Goodfellow et al., 2016, Section 19.5].

The difficulty is we need to sample from the model, which is unnormalized, and MCMC algorithms like the Langevin Dynamic is utilized. Running MCMC until convergence is computationally expensive. Methods like Contrastive Divergence (CD) [Hinton, 2002] are some alternative methods to approximate the gradient by some short run MCMC iterations.

2.1 Contrastive divergence

The main difficulty in ML training is in the negative phase, where samples from the model are needed. Hinton proposed the Contrastive Divergence method to use samples from $p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}) = \Pi^{(1)}_{\boldsymbol{\theta}} p_{\text{data}}(\mathbf{x})$, which stands for the distribution by running 1 step MCMC iteration over the samples from $p_{\text{data}}(\mathbf{x})$. Thus the gradient in equation 2 over the training data becomes to:

$$\begin{aligned}
\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x})] &= \mathbb{E}_{p_{\text{data}}(\mathbf{x})} [-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})] + \mathbb{E}_{\tilde{\mathbf{x}} \sim p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})} [\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})] \\
&\approx \mathbb{E}_{p_{\text{data}}(\mathbf{x})} [-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})] + \mathbb{E}_{\tilde{\mathbf{x}} \sim p^{(1)}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})} [\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})] \,.
\end{aligned} \tag{3}$$

The mathematical motivation for such substitution is that the CD method is actually approximately minimizing the following objective:

$$D_{\text{KL}} \left[p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})\right] - D_{\text{KL}} \left[p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})\right], \tag{4}$$

and the gradient of $\boldsymbol{\theta}$ w.r.t. it is:

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}} \left[D_{\text{KL}} \left[p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})\right] - D_{\text{KL}} \left[p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}') \parallel p_{\boldsymbol{\theta}}(\mathbf{x}')\right]\right] \\
&= -\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x})] - \underbrace{\nabla_{\boldsymbol{\theta}} p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}') \frac{\partial}{\partial p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}')} D_{\text{KL}} \left[p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}') \parallel p_{\boldsymbol{\theta}}(\mathbf{x}')\right]}_{\textcircled{1}} - \underbrace{\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{x}') \frac{\partial}{\partial p_{\boldsymbol{\theta}}(\mathbf{x}')} D_{\text{KL}} \left[p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}') \parallel p_{\boldsymbol{\theta}}(\mathbf{x}')\right]}_{\textcircled{2}},
\end{aligned} \tag{5}$$

let's examine ②:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{x}') \frac{\partial}{\partial p_{\boldsymbol{\theta}}(\mathbf{x}')} D_{\mathrm{KL}}\left[p_{\boldsymbol{\theta}}^{(1)}(\mathbf{x}') \parallel p_{\boldsymbol{\theta}}(\mathbf{x}')\right] &= \nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{x}') \int p_{\boldsymbol{\theta}}^{(1)}(\mathbf{x}') \frac{\partial}{\partial p_{\boldsymbol{\theta}}(\mathbf{x}')} \left(-\log p_{\boldsymbol{\theta}}(\mathbf{x}')\right) \mathrm{d}\mathbf{x}' \\
&= -\int p_{\boldsymbol{\theta}}^{(1)}(\mathbf{x}') \frac{\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{x}')}{p_{\boldsymbol{\theta}}(\mathbf{x}')} \mathrm{d}\mathbf{x}' \\
&= -\int p_{\boldsymbol{\theta}}^{(1)}(\mathbf{x}') \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}') \mathrm{d}\mathbf{x}' \\
&= -\mathbb{E}_{p_{\boldsymbol{\theta}}^{(1)}(\mathbf{x}')}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}')\right].
\end{aligned} \tag{6}$$

And we can see if we simply drop ①, then equation 5 becomes to:

$$-\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x})\right] + \mathbb{E}_{p_{\boldsymbol{\theta}}^{(1)}(\mathbf{x}')}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}')\right] = \mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}\left[\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})\right] + \mathbb{E}_{p_{\boldsymbol{\theta}}^{(1)}(\mathbf{x}')}\left[-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}')\right], \tag{7}$$

which is just the negative of equation 3. Although the discarded ① makes CD a biased algorithm, the bias is always small. Recently improved CD [Du et al., 2020] takes this term into consideration and makes the training procedure more stable.

3 Training with score matching

The key observation is the score of the EBM $\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) = -\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}}(\mathbf{x})$ is independent of the partition function $Z_{\boldsymbol{\theta}}$. Thus when we try to minimize the Fisher divergence between our data and the EBM, we can avoid dealing with the intractable parition $Z_{\boldsymbol{\theta}}$:

$$D_{\mathrm{F}}(p_{\mathrm{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})) = \mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}\left[\frac{1}{2}\left\|\nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})\right\|^2\right]. \tag{8}$$

3.1 Basic score matching

The problem is that $\nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x})$ is unknown. However, with integration by parts, the second derivatives of $E_{\boldsymbol{\theta}}(\mathbf{x})$ can be used to replace the unknown $\nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x})$ [Hyvärinen and Dayan, 2005].

$$\begin{aligned}
\mathcal{J}(\boldsymbol{\theta}) &= \mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}\left[\frac{1}{2}\left\|\nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})\right\|^2\right] \\
&= \int p_{\mathrm{data}}(\mathbf{x})\left[\frac{1}{2}\|\nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x})\|^2 + \frac{1}{2}\|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})\|^2 - \nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x})^\top \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})\right] \mathrm{d}\mathbf{x} \\
&= \int p_{\mathrm{data}}(\mathbf{x})\left[\frac{1}{2}\|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})\|^2\right] \mathrm{d}\mathbf{x} - \underbrace{\int p_{\mathrm{data}}(\mathbf{x})\left[\nabla_{\mathbf{x}} \log p_{\mathrm{data}}(\mathbf{x})^\top \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})\right] \mathrm{d}\mathbf{x}}_{①} + \text{const},
\end{aligned} \tag{9}$$

the computational difficulty exists in term ①, and we can conquer that with integration by parts, in the following we use d to denote the data dimensionality:

$$\begin{aligned}
① &= \sum_{i=1}^{d} \int p_{\mathrm{data}}(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_i} \log p_{\mathrm{data}}(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_i} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} \\
&= \sum_{i=1}^{d} \int \frac{\partial}{\partial \mathbf{x}_i} p_{\mathrm{data}}(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_i} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x},
\end{aligned} \tag{10}$$

and without loss of generality, we can examine the first term in the summation:

$$\begin{aligned}
&\int \frac{\partial}{\partial \mathbf{x}_1} p_{\text{data}}(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_1} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} \\
&= \int \left(\int \frac{\partial}{\partial \mathbf{x}_1} p_{\text{data}}(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_1} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x}_1 \right) \mathrm{d}\mathbf{x}_2 \cdots \mathbf{x}_d \\
&= \int \left[f(\mathbf{x}_{2:d}) - \int p_{\text{data}}(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_1} \frac{\partial}{\partial \mathbf{x}_1} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x}_1 \right] \mathrm{d}\mathbf{x}_2 \cdots \mathbf{x}_d \\
&= - \int p_{\text{data}}(\mathbf{x}) \frac{\partial^2}{\partial \mathbf{x}_1^2} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x},
\end{aligned} \tag{11}$$

where

$$f(\mathbf{x}_{2:d}) = \lim_{a \to \infty, b \to -\infty} \left(p_{\text{data}}(a, \mathbf{x}_{2:d}) \frac{\partial}{\partial \mathbf{x}_1} \log p_{\boldsymbol{\theta}}(a, \mathbf{x}_{2:d}) - p_{\text{data}}(b, \mathbf{x}_{2:d}) \frac{\partial}{\partial \mathbf{x}_1} \log p_{\boldsymbol{\theta}}(b, \mathbf{x}_{2:d}) \right),$$

and this term is assumed to be zero since the regularity condition of our model is: (1) $p_{\text{data}}(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})$ goes to zero for any $\boldsymbol{\theta}$ when $\|\mathbf{x}\| \to \infty$, (2) $p_{\text{data}}(\mathbf{x})$ is differentiable, (3) $\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})\|^2 \right]$ and $\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})\|^2 \right]$ are finite for every $\boldsymbol{\theta}$. The third line of equation 11 comes from integration by parts, which can move the $\frac{\partial}{\partial \mathbf{x}_1}$ from $p_{\text{data}}(\mathbf{x})$ to $\frac{\partial}{\partial \mathbf{x}_1} \log p_{\boldsymbol{\theta}}(\mathbf{x})$.

With the above derivation, we have:

$$\begin{aligned}
\mathcal{J}(\boldsymbol{\theta}) &= \int p_{\text{data}}(\mathbf{x}) \left[\frac{1}{2} \|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})\|^2 \right] \mathrm{d}\mathbf{x} + \sum_{i=1}^{d} \int p_{\text{data}}(\mathbf{x}) \frac{\partial^2}{\partial \mathbf{x}_1^2} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} + \text{const} \\
&= \int p_{\text{data}}(\mathbf{x}) \sum_{i=1}^{d} \left(\frac{1}{2} \left(\frac{\partial}{\partial \mathbf{x}_i} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \right)^2 + \frac{\partial^2}{\partial \mathbf{x}_i^2} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \right) \mathrm{d}\mathbf{x} + \text{const} \\
&= \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\sum_{i=1}^{d} \frac{1}{2} \left(\frac{\partial}{\partial \mathbf{x}_i} E_{\boldsymbol{\theta}}(\mathbf{x}) \right)^2 - \frac{\partial^2}{\partial \mathbf{x}_i^2} E_{\boldsymbol{\theta}}(\mathbf{x}) \right] + \text{const.}
\end{aligned} \tag{12}$$

In this way, we can learn the EBM when we can compute the score and Hessian of the energy function $E_{\boldsymbol{\theta}}(\mathbf{x})$.

3.2 Denoising score matching

There two main shortcomings of basic SM. First, it is only applicable to continuous and unbounded data, which cannot be used to digital data. Second is the computational cost of Hessian is $\mathcal{O}(d)$, and can not be applied to high dimensional data. One way to alleviate the problem is to add some smooth noise $\boldsymbol{\epsilon}$ to the data, and the resulting noisy distribution is $q(\tilde{\mathbf{x}}) = \int q(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})\mathrm{d}\mathbf{x}$. The interesting is:

$$\begin{aligned}
D_{\text{F}}(q(\tilde{\mathbf{x}}) \parallel p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})) &= \mathbb{E}_{q(\tilde{\mathbf{x}})}\left[\frac{1}{2}\|\nabla_{\tilde{\mathbf{x}}}\log q(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\|^2\right] \\
&= \mathbb{E}_{q(\tilde{\mathbf{x}})}\left[\frac{1}{2}\|\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\|^2\right] - \mathbb{E}_{q(\tilde{\mathbf{x}})}\left[\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})^\top\nabla_{\tilde{\mathbf{x}}}\log q(\tilde{\mathbf{x}})\right] + \text{const} \\
&= \mathbb{E}_{q(\tilde{\mathbf{x}})}\left[\frac{1}{2}\|\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\|^2\right] - \int\frac{q(\tilde{\mathbf{x}})}{q(\tilde{\mathbf{x}})}\left(\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})^\top\nabla_{\tilde{\mathbf{x}}}q(\tilde{\mathbf{x}})\right)\mathrm{d}\tilde{\mathbf{x}} + \text{const} \\
&= \mathbb{E}_{q(\tilde{\mathbf{x}})}\left[\frac{1}{2}\|\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\|^2\right] - \int\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})^\top\int\nabla_{\tilde{\mathbf{x}}}q(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})\mathrm{d}\mathbf{x}\mathrm{d}\tilde{\mathbf{x}} + \text{const} \\
&= \mathbb{E}_{q(\tilde{\mathbf{x}})}\left[\frac{1}{2}\|\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\|^2\right] - \int\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})^\top\int q(\tilde{\mathbf{x}}|\mathbf{x})\nabla_{\tilde{\mathbf{x}}}\log q(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})\mathrm{d}\mathbf{x}\mathrm{d}\tilde{\mathbf{x}} + \text{const} \\
&= \mathbb{E}_{q(\tilde{\mathbf{x}})}\left[\frac{1}{2}\|\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\|^2\right] - \int\int q(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})^\top\nabla_{\tilde{\mathbf{x}}}\log q(\tilde{\mathbf{x}}|\mathbf{x})\mathrm{d}\mathbf{x}\mathrm{d}\tilde{\mathbf{x}} + \text{const} \\
&= \mathbb{E}_{q(\tilde{\mathbf{x}})}\left[\frac{1}{2}\|\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\|^2\right] - \mathbb{E}_{q(\tilde{\mathbf{x}},\mathbf{x})}\left[\nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})^\top\nabla_{\tilde{\mathbf{x}}}\log q(\tilde{\mathbf{x}}|\mathbf{x})\right] + \text{const} \\
&= \mathbb{E}_{q(\tilde{\mathbf{x}},\mathbf{x})}\left[\frac{1}{2}\left\|\nabla_{\tilde{\mathbf{x}}}\log q(\tilde{\mathbf{x}}|\mathbf{x}) - \nabla_{\tilde{\mathbf{x}}}\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\right\|_2^2\right] + \text{const},
\end{aligned} \tag{13}$$

we can see in the last line of equation 13, both $p_{\text{data}}(\mathbf{x})$ and the second derivative of $p_{\boldsymbol{\theta}}(\mathbf{x})$ are avoided. The underlying intuition is that following the gradient of $\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})$ at some corrupted point $\tilde{\mathbf{x}}$ should ideally move us towards the clean sample $\mathbf{x}$ [Vincent, 2011]. One thing should be kept in mind is the learned score corresponds to the noisy data distribution $q(\tilde{\mathbf{x}})$ rather than the original noise-free $p_{\text{data}}(\mathbf{x})$, which makes DSM not a consistent estimator of $p_{\text{data}}(\mathbf{x})$. But we can attenuate the inconsistency to choose small noise level to make $q(\tilde{\mathbf{x}}) \approx p_{\text{data}}(\mathbf{x})$.

3.3 Sliced score matching

[TODO]

4 Contrastive divergence and score matching

Surprisingly, there exist some close connections [Hyvarinen, 2007] between Contrastive Divergence (section 2.1) and Score Matching (section 3.1). To see this, suppose we only run one step Langevin MCMC for CD, that is:

$$\mathbf{x}'(\boldsymbol{\theta}_s) = \mathbf{x} - \frac{\epsilon^2}{2}\nabla_{\mathbf{x}}E_{\boldsymbol{\theta}_s}(\mathbf{x}) + \epsilon\mathbf{z}, \quad \mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \mathbf{z} \sim \mathcal{N}(\mathbf{z}; 0, I), \tag{14}$$

here $\boldsymbol{\theta}_s$ denotes the current state of $\boldsymbol{\theta}$, and each $\mathbf{x}'$ depends on $\boldsymbol{\theta}_s$. The Taylor series expansion of $E_{\boldsymbol{\theta}}(\mathbf{x}')$ at $\mathbf{x}$ is given by:

$$E_{\boldsymbol{\theta}}(\mathbf{x}') = E_{\boldsymbol{\theta}}(\mathbf{x}) + \nabla_{\mathbf{x}}^\top E_{\boldsymbol{\theta}}(\mathbf{x})(\mathbf{x}' - \mathbf{x}) + \frac{1}{2}(\mathbf{x}' - \mathbf{x})^\top\nabla_{\mathbf{x}}^2 E_{\boldsymbol{\theta}}(\mathbf{x})(\mathbf{x}' - \mathbf{x}) + o(\epsilon^2), \tag{15}$$

the corresponding CD gradient is given in the second line of equation 3:

$$\begin{aligned} &\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})\right] + \mathbb{E}_{\mathbf{x}' \sim p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}')}\left[\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}')\right] \\ =& \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})\right] + \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{z};0,I)}\left[\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}\left(\mathbf{x} - \frac{\epsilon^2}{2}\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}_s}(\mathbf{x}) + \epsilon\mathbf{z}\right)\right]. \end{aligned} \tag{16}$$

From equation 15, we can further have:

$$\begin{aligned} &- E_{\boldsymbol{\theta}}(\mathbf{x}) + \mathbb{E}_{\mathbf{x}' \sim p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}')}\left[E_{\boldsymbol{\theta}}(\mathbf{x}')\right] \\ =& \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(0,I)}\left[-E_{\boldsymbol{\theta}}(\mathbf{x}) + E_{\boldsymbol{\theta}}(\mathbf{x}')\right] \\ =& \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(0,I)}\left[\nabla_{\mathbf{x}}^{\top} E_{\boldsymbol{\theta}}(\mathbf{x})(\mathbf{x}' - \mathbf{x}) + \frac{1}{2}(\mathbf{x}' - \mathbf{x})^{\top}\nabla_{\mathbf{x}}^2 E_{\boldsymbol{\theta}}(\mathbf{x})(\mathbf{x}' - \mathbf{x}) + o(\epsilon^2)\right] \\ =& \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(0,I)}\left[\nabla_{\mathbf{x}}^{\top} E_{\boldsymbol{\theta}}(\mathbf{x}) \underbrace{\left(-\frac{\epsilon^2}{2}\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}_s}(\mathbf{x}) + \epsilon\mathbf{z}\right)}_{\mathsf{A}} + \frac{1}{2}\mathsf{A}^{\top}\nabla_{\mathbf{x}}^2 E_{\boldsymbol{\theta}}(\mathbf{x})\mathsf{A} + o(\epsilon^2)\right] \\ =& -\frac{\epsilon^2}{2}\nabla_{\mathbf{x}}^{\top} E_{\boldsymbol{\theta}}(\mathbf{x})\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}_s}(\mathbf{x}) + \frac{\epsilon^2}{2}\mathrm{trace}(\nabla_{\mathbf{x}}^2 E_{\boldsymbol{\theta}}(\mathbf{x})) + o(\epsilon^2) \\ \approx& \frac{\epsilon^2}{2}\left(-\nabla_{\mathbf{x}}^{\top} E_{\boldsymbol{\theta}}(\mathbf{x})\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}_s}(\mathbf{x}) + \mathrm{trace}(\nabla_{\mathbf{x}}^2 E_{\boldsymbol{\theta}}(\mathbf{x}))\right) \\ =& -\frac{\epsilon^2}{2}\left(\nabla_{\mathbf{x}}^{\top}\log p_{\boldsymbol{\theta}}(\mathbf{x})\nabla_{\mathbf{x}}\log p_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}}^2\log p_{\boldsymbol{\theta}}(\mathbf{x})\right), \end{aligned} \tag{17}$$

we can further denote:

$$\begin{aligned} \mathcal{J}_{\mathsf{CD}}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\theta}_s) &= \frac{\epsilon^2}{2}\left(-\nabla_{\mathbf{x}}^{\top} E_{\boldsymbol{\theta}}(\mathbf{x})\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}_s}(\mathbf{x}) + \mathrm{trace}(\nabla_{\mathbf{x}}^2 E_{\boldsymbol{\theta}}(\mathbf{x}))\right) \\ &= \frac{\epsilon^2}{2}\left(\sum_{i=1}^{d} -\frac{\partial}{\partial \mathbf{x}_i}E_{\boldsymbol{\theta}}(\mathbf{x})\frac{\partial}{\partial \mathbf{x}_i}E_{\boldsymbol{\theta}_s}(\mathbf{x}) + \frac{\partial^2}{\partial \mathbf{x}_i^2}E_{\boldsymbol{\theta}}(\mathbf{x})\right), \end{aligned} \tag{18}$$

notice we always take $\boldsymbol{\theta}_s$ as a constant, then:

$$\frac{\partial}{\partial \boldsymbol{\theta}_k}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\mathcal{J}_{\mathsf{CD}}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\theta}_s)\right] = \frac{\epsilon^2}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\sum_{i=1}^{d} -\frac{\partial}{\partial \boldsymbol{\theta}_k}\frac{\partial}{\partial \mathbf{x}_i}E_{\boldsymbol{\theta}}(\mathbf{x})\frac{\partial}{\partial \mathbf{x}_i}E_{\boldsymbol{\theta}_s}(\mathbf{x}) + \frac{\partial}{\partial \boldsymbol{\theta}_k}\frac{\partial^2}{\partial \mathbf{x}_i^2}E_{\boldsymbol{\theta}}(\mathbf{x})\right], \tag{19}$$

and if we examine the gradient w.r.t. $\boldsymbol{\theta}$ of score matching (equation 12), we have:

$$\begin{aligned} &\frac{\partial}{\partial \boldsymbol{\theta}_k}\mathcal{J}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}_k}D_{\mathrm{F}}(p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})) \\ =& \frac{\partial}{\partial \boldsymbol{\theta}_k}\left(\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\sum_{i=1}^{d}\frac{1}{2}\left(\frac{\partial}{\partial \mathbf{x}_i}E_{\boldsymbol{\theta}}(\mathbf{x})\right)^2 - \frac{\partial^2}{\partial \mathbf{x}_i^2}E_{\boldsymbol{\theta}}(\mathbf{x})\right] + \mathsf{const}\right) \\ =& \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\sum_{i=1}^{d}\frac{\partial}{\partial \boldsymbol{\theta}_k}\frac{\partial}{\partial \mathbf{x}_i}E_{\boldsymbol{\theta}}(\mathbf{x})\frac{\partial}{\partial \mathbf{x}_i}E_{\boldsymbol{\theta}}(\mathbf{x}) - \frac{\partial}{\partial \boldsymbol{\theta}_k}\frac{\partial^2}{\partial \mathbf{x}_i^2}E_{\boldsymbol{\theta}}(\mathbf{x})\right]. \end{aligned} \tag{20}$$

From the above and equation 5, we can conclude that:

$$\begin{aligned}\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}\left[\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] &\approx \mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}\left[-\nabla_{\boldsymbol{\theta}}E_{\boldsymbol{\theta}}(\mathbf{x})\right] + \mathbb{E}_{\mathbf{x}'\sim p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}')}\left[\nabla_{\boldsymbol{\theta}}E_{\boldsymbol{\theta}}(\mathbf{x}')\right] \\ &\approx -\nabla_{\boldsymbol{\theta}}\left[D_{\mathrm{KL}}\left[p_{\mathrm{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})\right] - D_{\mathrm{KL}}\left[p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}') \parallel p_{\boldsymbol{\theta}}(\mathbf{x}')\right]\right] \\ &\approx -\frac{\epsilon^2}{2}\nabla_{\boldsymbol{\theta}}D_{\mathrm{F}}(p_{\mathrm{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})),\end{aligned} \tag{21}$$

the approximation in the first line comes from using one step Langevin MCMC instead of infinite many, the one in the second line comes from dropping ① in equation 5, and the one in the third line comes from dropping $o(\epsilon^2)$ from Taylor series expansion.
What equation 21 tells us is that every time we add $\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}\left[-\nabla_{\boldsymbol{\theta}}E_{\boldsymbol{\theta}}(\mathbf{x})\right] + \mathbb{E}_{\mathbf{x}'\sim p^{(1)}_{\boldsymbol{\theta}}(\mathbf{x}')}\left[\nabla_{\boldsymbol{\theta}}E_{\boldsymbol{\theta}}(\mathbf{x}')\right]$ multiplied by a stepsize to the current $\boldsymbol{\theta}$ to do approximate MLE, we are implicitly and approximately doing gradient descent of the Fisher divergence between $p_{\mathrm{data}}(\mathbf{x})$ and our model $p_{\boldsymbol{\theta}}(\mathbf{x})$.

4.1 Training with Noise Contrastive Estimation

(This closely related to density ratio estimation via binary classification.)
In NCE we introduce a noise distribution $p_n(\mathbf{x})$ which we can sample from and evaluate density, for example we can choose $p_n(\mathbf{x}) = \mathcal{N}(\mathbf{x}; 0, I)$. We also introduce a pseudo binary label y which is 1 if the point is from $p_{\mathrm{data}}(\mathbf{x})$ and is 0 if the point is from $p_n(\mathbf{x})$. Then we can define a mixture distribution of samples from p_{data} and p_n:

$$p_{n,\mathrm{data}}(\mathbf{x}) = p(y=1)p_{\mathrm{data}}(\mathbf{x}) + p(y=0)p_n(\mathbf{x}),$$

then the posterior of $y = 0$ is given by:

$$\begin{aligned}p_{n,\mathrm{data}}(y=0 \mid \mathbf{x}) &= \frac{p_{n,\mathrm{data}}(\mathbf{x} \mid y=0)p(y=0)}{p_{n,\mathrm{data}}(\mathbf{x})} \\ &= \frac{p_n(\mathbf{x})p(y=0)}{p(y=1)p_{\mathrm{data}}(\mathbf{x}) + p(y=0)p_n(\mathbf{x})} \\ &= \frac{p_n(\mathbf{x})}{\nu p_{\mathrm{data}}(\mathbf{x}) + p_n(\mathbf{x})},\end{aligned} \tag{22}$$

where $\nu = p(y=1)/p(y=0)$. Similarly we can define a mixture distribution of p_{data} and $p_{\boldsymbol{\theta}}$ and the posterior of $y = 0$ is given by:

$$p_{n,\boldsymbol{\theta}}(y=0 \mid \mathbf{x}) = \frac{p_n(\mathbf{x})}{\nu p_{\boldsymbol{\theta}}(\mathbf{x}) + p_n(\mathbf{x})}, \tag{23}$$

In NCE, we indirectly fit $p_{\boldsymbol{\theta}}(\mathbf{x})$ to $p_{\mathrm{data}}(\mathbf{x})$ through maximizing:

$$\mathbb{E}_{p_{n,\mathrm{data}}(\mathbf{x},y)}\left[\log p_{n,\boldsymbol{\theta}}(y \mid \mathbf{x})\right], \tag{24}$$

where $E_{\boldsymbol{\theta}}(\mathbf{x})$ and $Z_{\boldsymbol{\theta}}$ are taken independently, in other words, there is no model restriction that $Z_{\boldsymbol{\theta}} = \int \exp\left\{-E_{\boldsymbol{\theta}}(\mathbf{x})\right\}\mathrm{d}\mathbf{x}$. When the classifier is powerful enough, the optimal $p_{n,\boldsymbol{\theta}^*}(y \mid \mathbf{x})$ will recover $p_{n,\mathrm{data}}(y \mid \mathbf{x})$ and $p_{\boldsymbol{\theta}^*}(\mathbf{x})$ will recover $p_{\mathrm{data}}(\mathbf{x})$. Also, NCE provides the normalizing constant of an Energy-Based Model as a by-product of its training procedure.

4.1.1 relation with GAN

The optimal classifier between two distinct distributions $p_1(\mathbf{x}), p_2(\mathbf{x})$, is given by:

$$D^*(\mathbf{x}) = \frac{p_1(\mathbf{x})}{p_1(\mathbf{x}) + p_2(\mathbf{x})} = \frac{1}{1 + \exp\left(-\log \frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right)} = \sigma(-\log \frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}).$$

Where $\sigma(\cdot)$ denotes the sigmoid function. In the above, we assume that the prior probability that $p(D = 1) = p(D = 2) = 0.5$. This optimal classifier assumes that we use logistic regression to train the classifier. In the reverse direction, we have that logistic regression gives us a method to estimate the density which transform the density (ratio) estimation problem into a classification problem.

4.2 Minimizing the Stein Discrepancy

The Stein Discrepancy is defined as:

$$\mathbb{S}\left(p \parallel q\right) = \sup_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \left[\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) f(\mathbf{x}) + \operatorname{trace}\left(\nabla_{\mathbf{x}} f(\mathbf{x})\right)\right], \tag{25}$$

note that training EBMs with Stein Discrepancy circumvents sampling from the EBM and only relies on the score of the model [Grathwohl et al., 2020].

4.2.1 Equivalence between Fisher divergence and learned Stein Discrepancy

[TODO]

References

G. Casella and R. Berger. Statistical inference. 2002.

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

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

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, 2020.

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

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

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

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

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