How To Train Your Energy-Based Model

NIKHIL KANDPAL

ADVISOR: COLIN RAFFEL

Overview

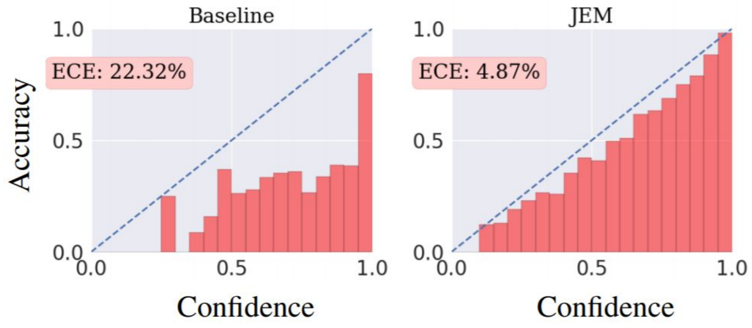
- Energy-based Model Introduction
- Method 1: MCMC-based Training
- Method 2: Learned Sampling Networks
- Method 3: Score Matching
- Future Directions

What is an Energy Based Model?

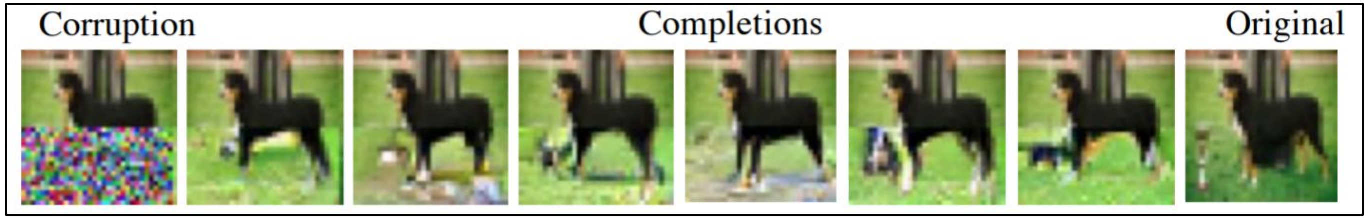
- EBM's are defined by an energy function $E_\theta : \mathbb{R}^d \rightarrow \mathbb{R}$
- $p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z(\theta)}$ where $Z(\theta) = \int e^{-E_\theta(x)} dx$
  - Low energy samples → high probability density
  - High energy samples → low probability density

Nijkamp et. al. 2019

Grathwohl et. al. 2020



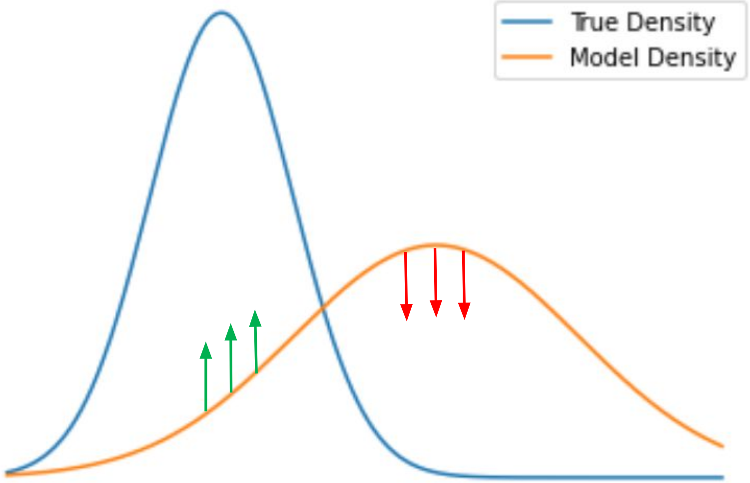
Du & Mordatch 2020



EBM Training

- Trained via Maximum Likelihood to match a target distribution p
- $L(\theta) = \mathbb{E}_{x \sim p}[-\log p_\theta(x)]$
- $\nabla_\theta L(\theta) = \underbrace{\mathbb{E}_{x^+ \sim p}[\nabla_\theta E_\theta(x^+)]}_{\text{Positive Phase}} - \underbrace{\mathbb{E}_{x^- \sim p_\theta}[\nabla_\theta E_\theta(x^-)]}_{\text{Negative Phase}}$

Independent of partition function $Z(\theta)$!



Overview

- Energy-based Model Introduction
- Method 1: MCMC-based Training
- Method 2: Learned Sampling Networks
- Method 3: Score Matching
- Future Directions

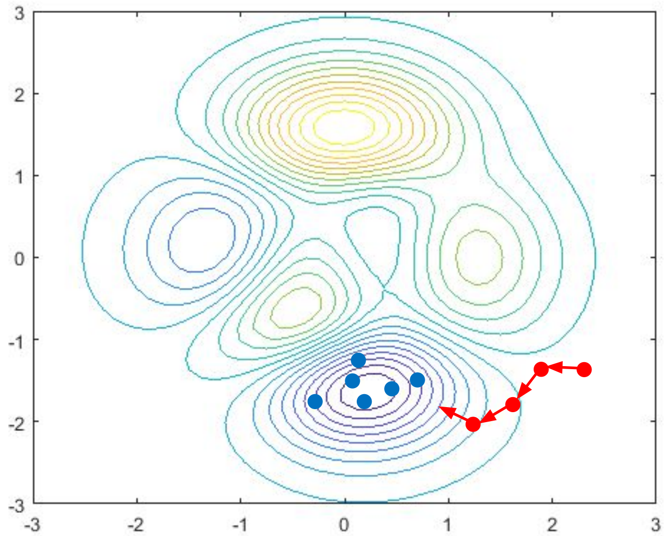
Markov Chain Monte Carlo-based Training

- $\nabla_\theta L(\theta) = \mathbb{E}_{x^+ \sim p}[\nabla_\theta E_\theta(x^+)] - \mathbb{E}_{x^- \sim p_\theta}[\nabla_\theta E_\theta(x^-)]$
- Problem: Evaluating $\nabla_\theta L(\theta)$ requires sampling from p_θ
- Solution: Sample using Markov Chain Monte Carlo (MCMC) during training

1. Initialize a chain by randomly initializing sample x_0
2. From a proposal distribution q, sample $x_{t+1} \sim q(\cdot \mid x_t)$
3. Accept the new sample with probability $e^{-E_\theta(x_{t+1}) + E_\theta(x_t)}$
4. Go to step 2

Stochastic Gradient Langevin Dynamics (SGLD)

- MCMC variant with gradient-based proposal distribution and no accept-reject step
- $x_{t+1} = x_t \underbrace{- \eta \nabla_x E_\theta(x_t)}_{\text{Gradient-based proposal efficiently finds low-energy regions}} + 2\sqrt{\eta}\omega\,,\quad \omega \sim N(0, I)\,,\ \underbrace{\eta \to 0 \text{ as } t \to \infty}_{\text{Decaying step size makes accept-reject unnecessary}}$



Intuition:

- Early trajectory behaves like gradient descent
- Late trajectory behaves like a random walk in the low-energy region

SGLD Training In Practice

- Separately select SGLD step size and noise scale hyperparameters
  - $x_{t+1} = x_t - \eta \nabla_x E_\theta(x_t) + \sigma\omega\,, \quad \omega \sim N(0, I)$
  - Note: This is equivalent to SGLD on a scaled version of E_θ (temperature sharpened distribution)
- Keep past samples in a buffer and sample from the buffer to initialize the SGLD chains

At each training iteration:

1. Initialize SGLD chains with samples from sample buffer
2. Run SGLD for a fixed number of steps to generate new samples
3. Update energy function parameters θ using generated samples and true samples
4. Store generated samples in sample buffer

Overview

- Energy-based Model Introduction
- Method 1: MCMC-based Training
- Method 2: Learned Sampling Networks
- Method 3: Score Matching
- Future Directions

Learned Sampling Networks

- Motivation: MCMC sampling is inherently sequential (i.e., slow)
  - Instead learn a function G that produces samples from p_θ
- Train E_θ using $L_E = \mathbb{E}_{x^+\sim p}[E_\theta(x^+)] - \mathbb{E}_{x^-\sim p_G}[E_\theta(x^-)]$ assuming $p_G \approx p_\theta$
- Train G using $L_G = KL(p_G||p_\theta) = \mathbb{E}_{x\sim p_G}[E_\theta(x)] - \mathcal{H}(p_G) + \log Z(\theta)$

Learned sampling methods vary in
how they maximize generator entropy

Examples From the Literature

- *Deep Directed Generative Models with Energy-Based Probability Estimation* (Kim & Bengio 2016)
  - Estimate generator entropy with layer activation entropy (assuming activations are normally distributed)

- *Maximum Entropy Generators for Energy-Based Models* (Kumar et. al. 2019)
  - Estimate generator entropy from mutual information between generator input and output

- *No MCMC for me: Amortized sampling for fast and stable training of energy-based models* (Grathwohl et. al. 2020)
  - Estimate generator entropy through a variational approximation

Overview

- Energy-based Model Introduction
- Method 1: MCMC-based Training
- Method 2: Learned Sampling Networks
- Method 3: Score Matching
- Future Directions

Score Matching

- Rather than match p_θ to p, instead match $\nabla_x \log p_\theta$ to $\nabla_x \log p$
- $L(\theta) = \frac{1}{2}\mathbb{E}_{x\sim p}[\|\nabla_x \log p_\theta\,(x) - \nabla_x \log p\,(x)\|_2^2]$

$$= \mathbb{E}_{x\sim p}\left[\frac{1}{2}\|\nabla_x \log p_\theta\,(x)\|_2^2 + tr(\nabla_x^2 \log p_\theta\,(x))\right] + \mathrm{C}$$

$$\propto \mathbb{E}_{x\sim p}\left[\frac{1}{2}\|\nabla_x E_\theta\,(x)\|_2^2 - tr(\nabla_x^2 E_\theta(x))\right] + \mathrm{C}$$

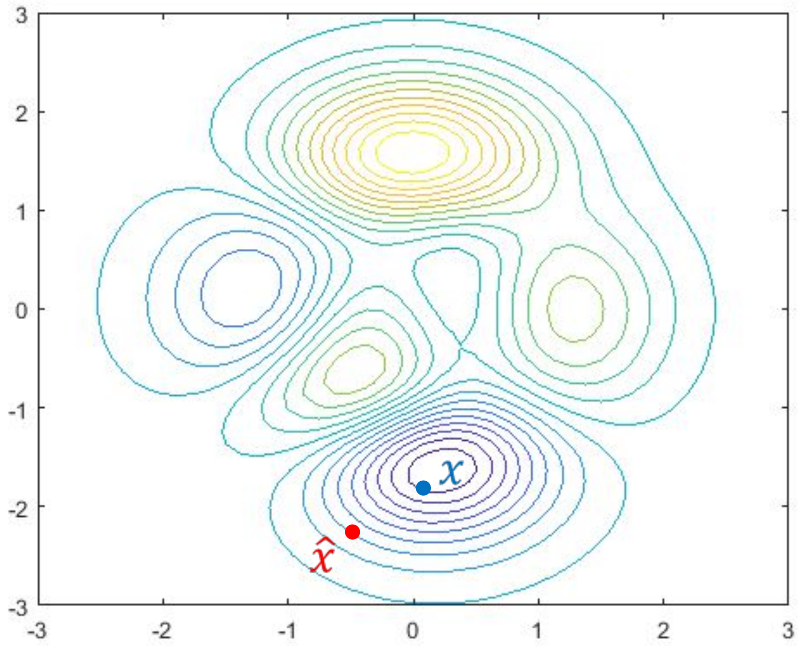
- Objective avoids computing partition function, but requires computing the Hessian trace

Sliced Score Matching

- Uses Hutchinson's Estimator: an efficient and unbiased estimator for matrix trace
- $tr(A) = \mathbb{E}[v^T A v]$ if $v \sim N(0, I)$
- Hessian trace can be computed as a Hessian-vector product
  - Avoids explicitly computing the full Hessian
  - Efficient with reverse-mode auto-differentiation (Pytorch, Tensorflow, etc.)

Denoising Score Matching

- The score matching objective can also be interpreted as denoising noisy samples
- $L(\theta) = \frac{1}{2}\mathbb{E}_{x\sim p,\, \hat{x}\sim N(x,\sigma^2)}\left[\left\| -\nabla_x E_\theta(\hat{x}) - \frac{x - \hat{x}}{\sigma^2} \right\|_2^2\right]$



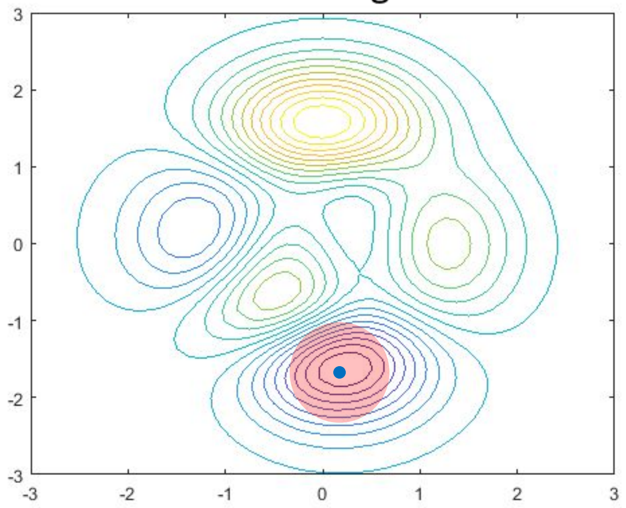
Intuition:

- $L(\theta)$ is minimized when $-\nabla_x E_\theta(\hat{x})$ points toward the original sample x
- Alternatively, a gradient descent step on E_θ starting from $\hat{x}$ should go in the direction of x

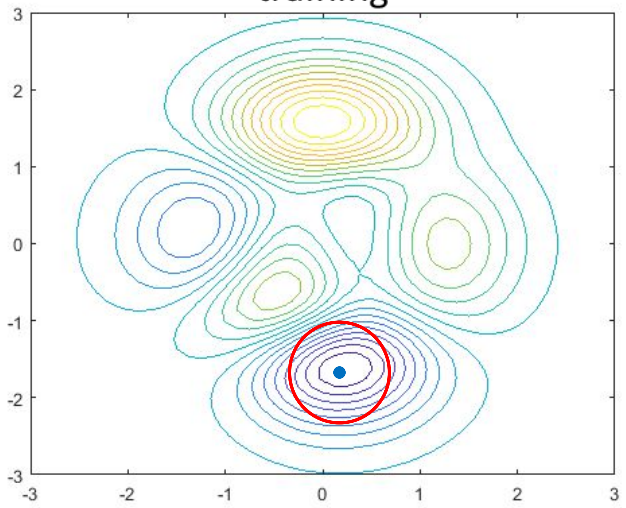
Rationale for Multiple Noise Scales

- Recent work suggests using multiple Gaussian noise scales $\{\sigma_i\}_{i=1}^{k}$
  - *Generative Modeling by Estimating Gradients of the Data Distribution* (Song & Ermon 2019)
  - *Learning Energy-Based Models in High-Dimensional Spaces with Multi-Scale Denoising Score Matching* (Li et. al. 2019)

The samples x and $\hat{x}$ we *think* are seen during training



The samples x and $\hat{x}$ that are *actually* seen during training



Comparison of Training Methods

	Strengths	Weaknesses
MCMC-Based Training	• Implicit sampling using backprop on the EBM provides a good inductive bias (convolutional EBM de-convolutional sampling)	• Slow due to sequential MCMC at each training iteration • Tricky to tune SGLD hyperparameters to get chains to mix quickly
Learned Samplers	• Efficient training • Efficient sampling	• Unstable training due to joint optimization of generator and EBM • Double the number of parameters to estimate
Score Matching	• Efficient and stable training	• Only trains on points in the vicinity of the true data distribution

Overview

- Energy-based Model Introduction
- Method 1: MCMC-based Training
- Method 2: Learned Sampling Networks
- Method 3: Score Matching
- Future Directions

Future Directions

- Energy functions are extremely flexible (any scalar function of the data)
- What values would make for interesting and useful energy functions?
  - Example: Classifier uncertainty as an energy function
  - Low uncertainty on the data distribution is a common inductive bias in semi-supervised learning
  - High uncertainty away from the data distribution is desirable, but not currently a feature of modern deep learning models
- More generally, how can EBM training be incorporated into classifiers to give classifiers useful properties?

References

- SGLD
  - Max Welling, Yee Whye Teh. *Bayesian Learning via Stochastic Gradient Langevin Dynamics*. 2011.
- MCMC-based Training
  - Erik Nijkamp, Mitch Hill, Song-Chun Zhu, Ying Nian Wu. *Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model.* 2019.
  - Yilun Du, Igor Mordatch. *Implicit Generation and Generalization in Energy-Based Models*. 2020.
  - Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, Kevin Swersky. *Your Classifier is Secretly an Energy Based Model and You Should Treat it Like One*. 2020.
- Learned Sampling Networks
  - Taesup Kim, Yoshua Bengio. *Deep Directed Generative Models with Energy-Based Probability Estimation*. 2016.
  - Rithesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, Yoshua Bengio. *Maximum Entropy Generators for Energy-Based Models*. 2019.
  - Will Grathwohl, Jacob Kelly, Milad Hashemi, Mohammad Norouzi, Kevin Swersky, David Duvenaud. *No MCMC for me: Amortized sampling for fast and stable training of energy-based models*. 2020.
- Score Matching
  - Aapo Hyvarinen. *Estimation of Non-Normalized Statistical Models by Score Matching*. 2005.
  - Pascal Vincent. *A Connection Between Score Matching and Denoising Autoencoders*. 2010.
  - Yang Song, Sahaj Garg, Jiaxin Shi, Stefano Ermon. *Sliced Score Matching: A Scalable Approach to Density and Score Estimation*. 2019.
  - Yang Song, Stefano Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*. 2019.
  - Zengyi Li, Yubei Chen, Friedrich T. Sommer. *Learning Energy-Based Models in High-Dimensional Spaces with Multi-scale Denoising Score Matching*. 2019.