Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model

Erik Nijkamp, Mitch Hill, Song-Chun Zhu, Ying Nian Wu
University of California, Los Angeles

Abstract

This paper studies a curious phenomenon in learning energy-based model (EBM) using MCMC. In each learning iteration, we generate synthesized examples by running a non-convergent, non-mixing, and non-persistent short-run MCMC toward the current model, always starting from the same initial distribution such as uniform noise distribution, and always running a fixed number of MCMC steps. After generating synthesized examples, we then update the model parameters according to the maximum likelihood learning gradient, as if the synthesized examples are fair samples from the current model. We treat this non-convergent short-run MCMC as a learned generator model or a flow model. We provide arguments for treating the learned non-convergent short-run MCMC as a valid model. We show that the learned short-run MCMC is capable of generating realistic images. More interestingly, unlike traditional EBM or MCMC, the learned short-run MCMC is capable of reconstructing observed images and interpolating between images, like generator or flow models.

Maximum Likelihood Learning of EBM

Probability Density

Let x be the signal, such as an image. The energy-based model (EBM) is a Gibbs distribution

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x)), \tag{1}$$

where we assume x is within a bounded range. $f_\theta(x)$ is the negative energy and is parametrized by a bottom-up convolutional neural network (ConvNet) with weights θ. $Z(\theta) = \int \exp(f_\theta(x))dx$ is the normalizing constant.

Analysis by Synthesis

Suppose we observe training examples $x_i, i = 1, ..., n \sim p_{\text{data}}$, where p_{data} is the data distribution. For large n, the sample average over $\{x_i\}$ approximates the expectation with respect with p_{data}.

The log-likelihood is

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i) \doteq \mathbb{E}_{p_{\text{data}}}[\log p_\theta(x)]. \tag{2}$$

The derivative of the log-likelihood is

$$\begin{aligned} L'(\theta) &= \mathbb{E}_{p_{\text{data}}}\left[\frac{\partial}{\partial\theta} f_\theta(x)\right] - \mathbb{E}_{p_\theta}\left[\frac{\partial}{\partial\theta} f_\theta(x)\right] \\ &\doteq \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\theta} f_\theta(x_i) - \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\theta} f_\theta(x_i^-), \end{aligned} \tag{3}$$

where $x_i^- \sim p_\theta(x)$ for $i = 1, ..., n$ are the generated examples from the current model $p_\theta(x)$.

The above equation leads to the "analysis by synthesis" learning algorithm. At iteration t, let θ_t be the current model parameters. We generate $x_i^- \sim p_{\theta_t}(x)$ for $i = 1, ..., n$. Then we update $\theta_{t+1} = \theta_t + \eta_t L'(\theta_t)$, where η_t is the learning rate.

Short-Run MCMC

Sampling by Langevin Dynamics

Generating synthesized examples $x_i^- \sim p_\theta(x)$ requires MCMC, such as Langevin dynamics, which iterates

$$x_{\tau+\Delta\tau} = x_\tau + \frac{\Delta\tau}{2} f'_\theta(x_\tau) + \sqrt{\Delta\tau}U_\tau, \tag{4}$$

where τ indexes the time, $\Delta\tau$ is the discretization of time, and $U_\tau \sim \text{N}(0, I)$ is the Gaussian noise term.

Guidance by Energy-based Model

If $f_\theta(x)$ is multi-modal, then different chains tend to get trapped in different local modes, and they do not mix. We propose to give up the sampling of p_θ. Instead, we run a fixed number, e.g., K, steps of MCMC, toward p_θ, starting from a fixed initial distribution, p_0, such as the uniform noise distribution. Let M_θ be the K-step MCMC transition kernel. Define

$$q_\theta(x) = (M_\theta p_0)(z) = \int p_0(z) M_\theta(x|z)dz, \tag{5}$$

which is the marginal distribution of the sample x after running K-step MCMC from p_0.

Instead of learning p_θ, we treat q_θ to be the target of learning. After learning, we keep q_θ, but we discard p_θ. That is, the sole purpose of p_θ is to guide a K-step MCMC from p_0.

Learning Short-Run MCMC

The learning algorithm is as follows. Initialize θ_0. At learning iteration t, let θ_t be the model parameters. We generate $x_i^- \sim q_{\theta_t}(x)$ for $i = 1, ..., m$. Then we update $\theta_{t+1} = \theta_t + \eta_t \Delta(\theta_t)$, where

$$\begin{aligned} \Delta(\theta) &= \mathbb{E}_{p_{\text{data}}}\left[\frac{\partial}{\partial\theta} f_\theta(x)\right] - \mathbb{E}_{q_\theta}\left[\frac{\partial}{\partial\theta} f_\theta(x)\right] \\ &\approx \sum_{i=1}^{m}\frac{\partial}{\partial\theta} f_\theta(x_i) - \sum_{i=1}^{m}\frac{\partial}{\partial\theta} f_\theta(x_i^-). \end{aligned} \tag{6}$$

The learning procedure is simple. The key to the algorithm is that the generated $\{x_i^-\}$ are independent and fair samples from the model q_θ.

Algorithm 1: Learning short-run MCMC.

```
input  : Negative energy $f_\theta(x)$, training steps $T$, initial weights $\theta_0$, observed
         examples $\{x_i\}_{i=1}^n$, batch size $m$, variance of noise $\sigma^2$, Langevin descretization $\Delta\tau$
         and steps $K$, learning rate $\eta$.
output : Weights $\theta_{T+1}$.
for $t = 0 : T$ do
    1. Draw observed images $\{x_i\}_{i=1}^m$.
    2. Draw initial negative examples $\{x_i^-\}_{i=1}^m \sim p_0$.
    3. Update observed examples $x_i \leftarrow x_i + \varepsilon_i$ where $\varepsilon_i \sim \text{N}(0, \sigma^2 I)$.
    4. Update negative examples $\{x_i^-\}_{i=1}^m$ for $K$ steps of Langevin dynamics (4).
    5. Update $\theta_t$ by $\theta_{t+1} = \theta_t + g(\Delta(\theta_t), \eta, t)$ where gradient $\Delta(\theta_t)$ is (6) and $g$ is ADAM.
```

Relation to Moment Matching Estimator

We may interpret Short-Run MCMC as Moment Matching Estimator. We outline the case of a learning the top-filters of a ConvNet:

- Consider $f_\theta(x) = \langle \theta, h(x) \rangle$ where $h(x)$ are the top-layer filter responses of a pretrained ConvNet with top-layer weights θ.
- For such $f_\theta(x)$, we have $\frac{\partial}{\partial\theta} f_\theta(x) = h(x)$.
- The MLE estimator of p_θ is a moment-matching estimator, i.e. $\mathbb{E}_{p_{\hat{\theta}_{\text{MLE}}}}[h(x)] = \mathbb{E}_{p_{\text{data}}}[h(x)]$.
- If we use the short-run MCMC learning algorithm, it will converge (assume convergence is attainable) to a moment matching estimator, i.e., $\mathbb{E}_{q_{\hat{\theta}_{\text{MME}}}}[h(x)] = \mathbb{E}_{p_{\text{data}}}[h(x)]$.
- Thus, the learned model $q_{\hat{\theta}_{\text{MME}}}(x)$ is a valid estimator in that it matches to the data distribution in terms of sufficient statistics defined by the EBM.

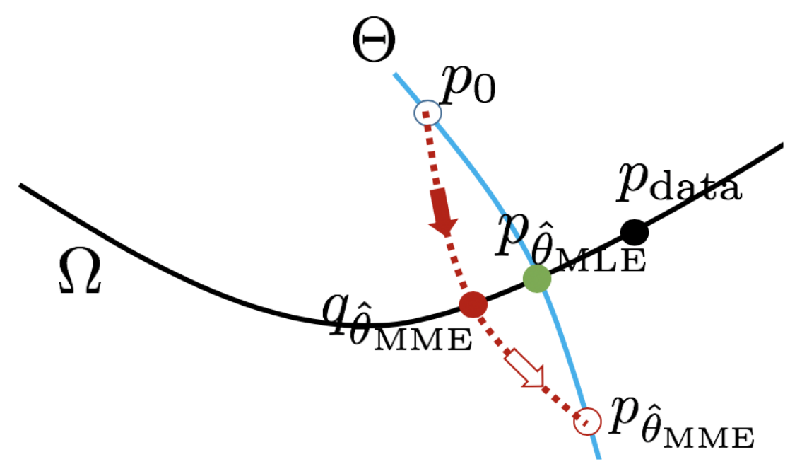
Θ
p_0
p_{data}
$p_{\hat{\theta}_{\text{MLE}}}$
Ω
$q_{\hat{\theta}_{\text{MME}}}$
$p_{\hat{\theta}_{\text{MME}}}$

Figure 1: The blue curve illustrates the model distributions corresponding to different values of parameter θ. The black curve illustrates all the distributions that match p_{data} (black dot) in terms of $\mathbb{E}[h(x)]$. The MLE $p_{\hat{\theta}_{\text{MLE}}}$ (green dot) is the intersection between Θ (blue curve) and Ω (black curve). The MCMC (red dotted line) starts from p_0 (hollow blue dot) and runs toward $p_{\hat{\theta}_{\text{MME}}}$ (hollow red dot), but the MCMC stops after K-step, reaching $q_{\hat{\theta}_{\text{MME}}}$ (red dot), which is the learned short-run MCMC.

Relation to Generator Model

We may consider $q_\theta(x)$ to be a generative model,

$$z \sim p_0(z);\ x = M_\theta(z, u), \tag{7}$$

where u denotes all the randomness in the short-run MCMC. For the K-step Langevin dynamics, M_θ can be considered a K-layer noise-injected residual network. z can be considered latent variables, and p_0 the prior distribution of z. Due to the non-convergence and non-mixing, x can be highly dependent on z, and z can be inferred from x.

Interpolation

We can perform interpolation as follows. Generate z_1 and z_2 from $p_0(z)$. Let $z_\rho = \rho z_1 + \sqrt{1-\rho^2} z_2$. This interpolation keeps the marginal variance of z_ρ fixed. Let $x_\rho = M_\theta(z_\rho)$. Then x_ρ is the interpolation of $x_1 = M_\theta(z_1)$ and $x_2 = M_\theta(z_2)$. Figure 3 displays x_ρ for a sequence of $\rho \in [0, 1]$.

Reconstruction

For an observed image x, we can reconstruct x by running gradient descent on the least squares loss function $L(z) = \|x - M_\theta(z)\|^2$, initializing from $z_0 \sim p_0(z)$, and iterates $z_{t+1} = z_t - \eta_t L'(z_t)$. Figure 4 displays the sequence of $x_t = M_\theta(z_t)$.

Capability 1: Synthesis

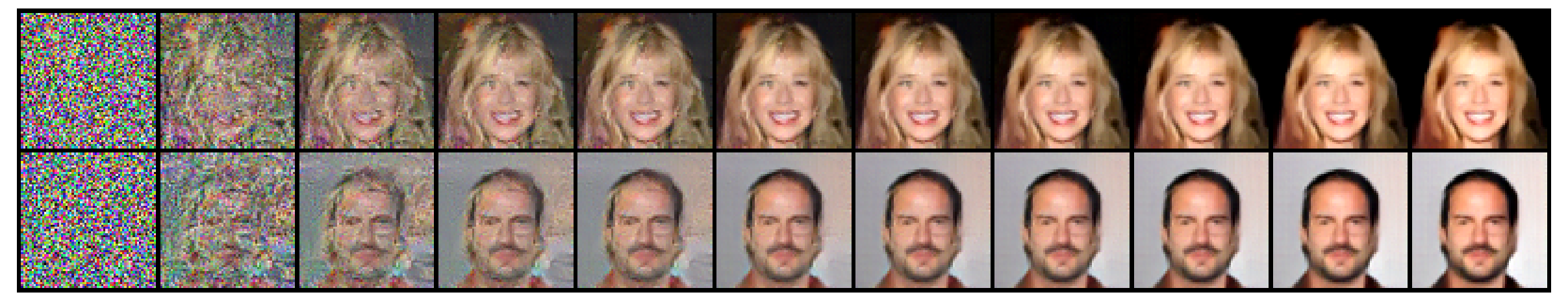

Figure 2: Generating synthesized examples by running 100 steps of Langevin dynamics initialized from uniform noise for CelebA (64 × 64).

Capability 2: Interpolation

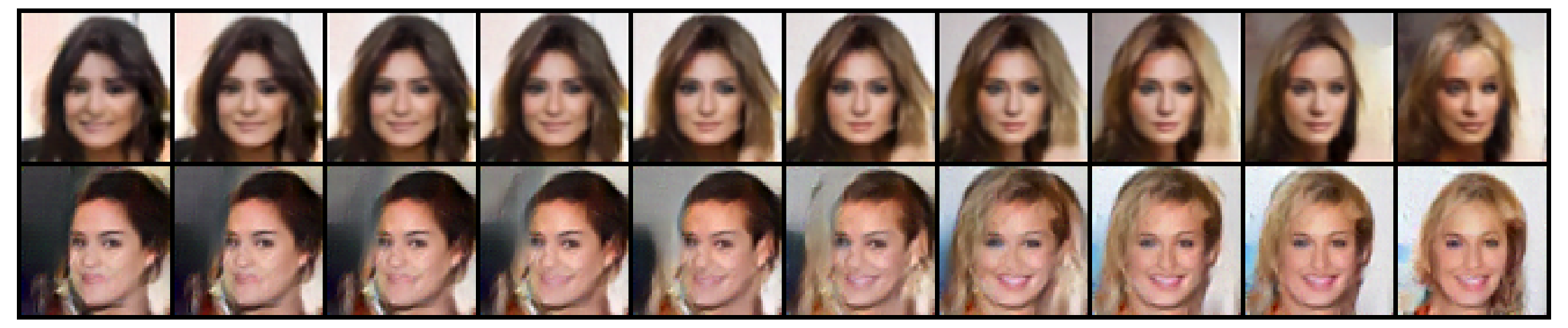

Figure 3: $M_\theta(z_\rho)$ with interpolated noise $z_\rho = \rho z_1 + \sqrt{1-\rho^2} z_2$ where $\rho \in [0, 1]$ on CelebA (64×64). Left: $M_\theta(z_1)$. Right: $M_\theta(z_2)$.

Capability 3: Reconstruction

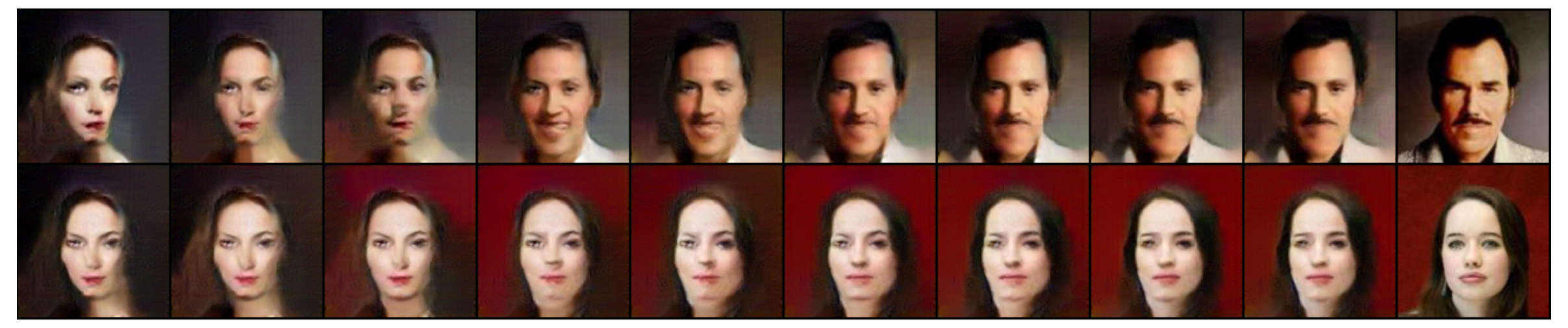

Figure 4: $M_\theta(z_t)$ over time t from random initialization $t = 0$ to reconstruction $t = 200$ on CelebA. Left: Random initialization. Right: Observed examples.

Conclusion

(1) We propose to shift the focus from convergent MCMC towards efficient, non-converging, non-mixing, short-run MCMC guided by EBM.

(2) We interpret short-run MCMC as Moment Matching Estimator and explore the relations to residual networks and generator-based models.

(3) We demonstrate the abilities of interpolation and reconstruction due to non-mixing MCMC, which goes far beyond the capacity of convergent MCMC.

References

- J Xie*, Y Lu*, SC Zhu, YN Wu. A Theory of Generative ConvNet, *ICML* 2016.
- R Gao*, Y Lu, J Zhou, SC Zhu, YN Wu. Learning generative ConvNets via Multigrid Modeling and Sampling, *CVPR* 2018.
- E Nijkamp*, M Hill*, SC Zhu, YN Wu. On the Anatomy of MCMC-based Maximum Likelihood Learning of Energy-Based Models, *AAAI* 2020.