Langevin Dynamics and Deep Neural Network

Kan Chen

Applied Math and Computational Science
University of Pennsylvania

December 5, 2019

Outline

1. Sampling via Langevin Dynamics.
2. Sampling and Non-convex Optimization.
3. Langevin Dynamics and DNN.

Outline

1. Sampling via Langevin Dynamics.
2. Sampling and Non-convex Optimization.
3. Langevin Dynamics and DNN.

Overview of Langevin Dynamics

In physics, Langevin dynamics is an approach to the mathematical modeling of the dynamics of molecular systems. It was originally developed by French physicist **Paul Langevin**.The approach is characterized by the use of simplified models while accounting for omitted degrees of freedom by the use of stochastic differential equations. (from *Wikipedia*)

Overview of Langevin Dynamics

Consider the following Stochastic Differential Equation:

$$d\mathbf{X}_t = -\nabla f(\mathbf{X}_t)dt + \sqrt{2}d\mathbf{W}_t \tag{1}$$

where $\{\mathbf{X}_t\}_t \in \mathbb{R}^p$ is a series of random variables, $\{\mathbf{W}_t\}_t$ is a series of standard Wiener process, and f is any smooth function. Then the stationary distribution of X_t when $t \to \infty$ is

$$p(\mathbf{x}) \propto e^{-f(\mathbf{x})}$$

Overview of Langevin Dynamics

The reason behind this is **Fokker-Planck** equation. Consider the general stochastic differential equation in this form:

$$d\mathbf{X}_t = \mu(\mathbf{X}_t, t)dt + \sigma(\mathbf{X}_t, t)d\mathbf{W}_t \tag{2}$$

where the drift term is $\mu(\mathbf{X}_t, t)$, diffusion coefficient is $D(\mathbf{X}_t, t) = \sigma^2(\mathbf{X}_t, t)/2$.

Overview of Langevin Dynamics

For 1-D case, the **Fokker-Planck** equation for probability density $p(\mathbf{x}, t)$ of random variable $\mathbf{X}_t$ is:

$$\frac{\partial}{\partial t} p(\mathbf{x}, t) = -\frac{\partial}{\partial x}[\mu(\mathbf{X}_t, t) p(\mathbf{x}, t)] + \frac{\partial^2}{\partial x^2}[D(\mathbf{X}_t, t) p(\mathbf{x}, t)] \tag{3}$$

Simple Proof

See the white board for details.

Sampling and Langevin Dynamics

Problem: given a high dimensional density $p(\mathbf{x})$, we want to generate a series of points $\{\mathbf{x}_i\}$ such that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{x}_i) = p(\mathbf{x})$$

Sampling and Langevin Dynamics

We can solve the following SDE

$$d\mathbf{X}_t = -\nabla f(\mathbf{X}_t)dt + \sqrt{2}d\mathbf{W}_t \tag{4}$$

by using the Euler discretization method:

$$\mathbf{X}_{t+1} - \mathbf{X}_t = -\nabla f(\mathbf{X}_t)h + \sqrt{2h}\xi_t \tag{5}$$

where ξ_t is a series of Gaussian vector and h is the step-size (learning rate).

Sampling and Langevin Dynamics

We can simultaneously solve N SDEs at the same time to get N points. Then when $t \to \infty$ and $N \to \infty$, we have:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{x}_i) = p(\mathbf{x}) \tag{6}$$

Compare with Stein's Variation Sampling

- **Langevin Dynamics**: solve the SDE by discretization and get $\mathbf{x}$. (Just get one point, get N points by solving N SDEs simultaneously)
- **Stein's variation**: generate a series of data points from distribution q, after enough iterations, the empirical distribution of these points will converge to target distribution p. (See white board for details)

ULA and MALA

- **ULA (Unadjusted Langevin Algorithm)**: directly solve the SDE without any adjustment.
- **MALA (Metropolis-adjusted Langevin Algorithm)**: envolve the reject and accept process. (See the white board for details)

Outline

Non-convex Optimization

- In DNN, the most fundamental thing is to minimize the cost function or maximize the utility function.
- For convex optimization, it is easy.
- However, in many cases, we have to face the non-convex optimization problem.

Non-convex Optimization

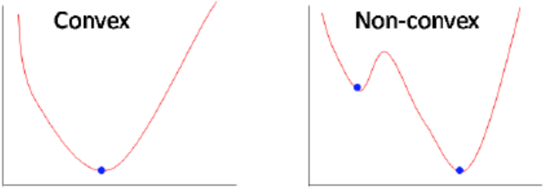

Figure: convex optimization v.s. non-convex optimization

Existing Method

- Gradient Descent?
- Often trapped by the saddle points or local minimums.
- Very sensitive to the choice of initial point.

Improvement

Simulated Annealing (SA): a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space for an optimization problem. (from *Wikipedia*)

Details about Simulated Annealing

See the white board for details.

Sampling and Optimization

- If we try to use gradient descent, then the choice of initial point is very important.
- If the objective function that we are trying to minimize is $f(x)$, then what would be the properties of $p(x) \propto e^{-f(x)}$?

Outline

Literature

- Langevin Dynamics with Continuous Tempering for Training Deep Neural Networks. *Nanyang Ye, Zhanxi Zhu and Rafal K.Mantiuk*
- Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. Chuanyuan Li, Changyou Chen, David Carlson and Lawrence Carin

Two Phases for Training Neural Network

- Sampling enough points to capture the modes before running optimization algorithm.
- Pick up a point near the the top mode as the initialization.

Two Phases for Training Neural Network

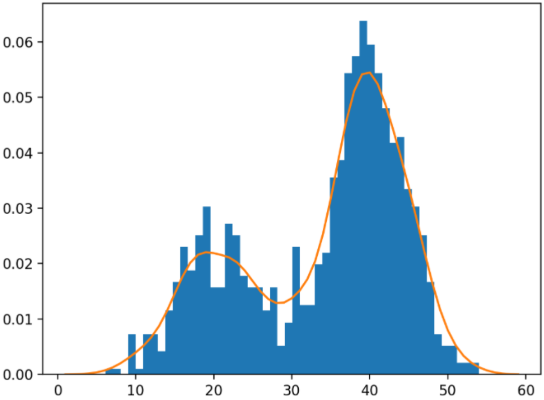

Figure: 1D case capture the modes

Combining SA and Langevin

Langevin Dynamics and SA:

$$dX_t = -\nabla f(X_t)dt + \sqrt{2\beta^{-1}(t)}dW_t \tag{7}$$

where $\beta^{-1}(t) = k_B T(t)$ decays as $T(t) = c/\log(2+t)$. The idea to very similar to MALA.

Insight

We can create a reversible Markov chain by adding the reject-accept process. This is just a very superficial part of Langevin Dynamics and sampling. (Overdamped Langevin)

Thank you!