Generative Modeling by Estimating Gradients of the Data Distribution

Yang Song
Stefano Ermon

Stanford AI Lab
Representations of Probability Distributions

**Implicit models**: directly represent the sampling process

\[ \epsilon \sim p(\epsilon) \]

\[ x = g(\epsilon) \]

**Cons**: hard to train, no likelihood, no principled model comparisons
Representations of Probability Distributions

**Explicit models**: represent a probability density/mass function $p(x)$

- **Cons**: need to be normalized $\Rightarrow$ balance expressivity and tractability

- Bayesian networks (e.g., VAEs)
- MRF
- Autoregressive models
- Flow models

Stanford University
Representation of Probability Distributions

This talk: The gradient of a probability density w.r.t. the input dimensions

$$\nabla_x \log p(x)$$

Score

NOT the gradient w.r.t. model parameters
Score Estimation

- **Given:** i.i.d. samples \( \{x_1, x_2, \cdots, x_N \} \sim p_{\text{data}}(x) \)
- **Task:** Estimating the score \( \nabla_x \log p_{\text{data}}(x) \)
- **Score Model:** A trainable vector-valued function \( s_{\theta}(x) : \mathbb{R}^D \rightarrow \mathbb{R}^D \)
- **Objective:** How to compare two vector fields of scores?

\[
\frac{1}{2} \mathbb{E}_{p_{\text{data}}} \left[ \| \nabla_x \log p_{\text{data}}(x) - s_{\theta}(x) \|^2 \right]
\]

(Fisher divergence)

- **Integration by parts**

\[
\begin{align*}
\mathbb{E}_{p_{\text{data}}} & \left[ \frac{1}{2} \| s_{\theta}(x) \|^2 + \text{trace}(\nabla_x s_{\theta}(x)) \right] \\
\approx & \frac{1}{N} \sum_{i=1}^{N} \left[ \frac{1}{2} \| s_{\theta}(x_i) \|^2 + \text{trace}(\nabla_x s_{\theta}(x_i)) \right]
\end{align*}
\]

Hyvärinen (2005)
From Scores to Samples: Langevin Dynamics

Scores

Follow the scores

Follow noisy scores: Langevin dynamics

\[
\begin{align*}
    \mathbf{x}_{t+1} &\leftarrow \mathbf{x}_t + \frac{\epsilon}{2} s_\theta(\mathbf{x}_t) \\
    \mathbf{z}_t &\sim \mathcal{N}(0, I) \\
    \mathbf{x}_{t+1} &\leftarrow \mathbf{x}_t + \frac{\epsilon}{2} s_\theta(\mathbf{x}_t) + \sqrt{\epsilon} \mathbf{z}_t
\end{align*}
\]
Score-Based Generative Modeling

\[ \{x_1, x_2, \ldots, x_N\} \overset{\text{i.i.d.}}{\sim} p_{\text{data}}(x) \]

Score Matching

Scores

\[ s_\theta(x) \approx \nabla_x \log p_{\text{data}}(x) \]

Langevin dynamics

New samples

Stanford University
Adding Noise to Data for Well-Defined Scores

- Scores can be undefined when
  - The support of data distribution is on a low-dimensional manifold
  - The data distribution is discrete
- Solution: adding noise

Data unperturbed

Data perturbed with $\mathcal{N}(0; 0.0001)$
Challenge in Low Data Density Regions

\[
\frac{1}{2} \mathbb{E}_{p_{\text{data}}} \| \nabla_x \log p_{\text{data}}(x) - s_\theta(x) \|_2^2 \approx \frac{1}{2N} \sum_{i=1}^{N} \| \nabla_x \log p_{\text{data}}(x_i) - s_\theta(x_i) \|_2^2
\]
Adding Noise to Data for Better Score Estimation

- Random noise provides samples in low data density regions.
Trading off Sample Quality and Estimation Accuracy

Worse sample quality!
Better score estimation!

(Red encodes error)
Joint Score Estimation via Noise Conditional Score Networks
Annealed Langevin Dynamics: Joint Scores to Samples

- Sample using $\sigma_1, \sigma_2, \ldots, \sigma_L$ sequentially with Langevin dynamics.
- Anneal down the noise level.
- Samples used as initialization for the next level.
Experiments: Sampling
## Experiments: Sample Quality

### CIFAR-10 Unconditional

| Model          | Inception Score (higher is better) | FID score (lower is better) |
|----------------|-----------------------------------|----------------------------|
| PixelCNN       | 4.60                              | 65.93                      |
| EBM            | 6.02                              | 40.58                      |
| SNGAN          | 8.22 ± 0.05                       | 21.7                       |
| ProgressiveGAN | 8.80 ± 0.05                       | -                          |
| **NCSN (ours)** | **8.87 ± 0.12**                   | **25.32**                  |
Experiments: Inpainting
Conclusion

• Score-based generative modeling
  • No need to be normalized / invertible
    • Flexible architecture choices
  • No minimax optimization
    • stable training
    • a natural measurement of training progress / model comparison

• Adding noise and annealing the noise levels are critical

• Better or comparable sample quality to GANs.
Related Work

• Generative Stochastic Networks (Bengio et al. (2013), Alain et al. (2016))
  • Sampling starts close to data points.
  • Need **MCMC during training** with walkback.

• Nonequilibrium Thermodynamics (Sohl-Dickstein et al. (2015)), Infusion Training (Bordes et al. (2017)), Variational Walkback (Goyal et al. (2017))
  • **Likelihood**-based training.
  • Need **MCMC during training**.
Future Directions

• How to apply score-based generative modeling to discrete data?

• Theoretical guidance on how to choose noise levels?

• Better architecture for higher resolution image generation?

• Improved score estimation?