Diffusion Models II

Score-based models

May 1st, 2024

Recap

- Introduced Denoising Diffusion Probabilistic Models (DDPMs)
  - Idea: add Gaussian noise and train a model to iteratively remove the noise
  - We saw that the simplified loss was equivalent to learning the underlying distribution. In particular, we learned the score of a distribution (the gradient of the
- We are going to see an alternative way to define diffusion models based now on explicitly learning the scores of distributions
- This will give us a way to generalize diffusion models into a single, flexible framework.

Why care about scores?

- If you have an energy-based model, if the energy is given by $E_\theta(\mathbf{x})$ then the probability distribution associated with it is given by

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z_\theta},$$

where the Z_θ normalizes the distribution to sum to 1.

- If one knows the energy, then sampling from $p_\theta(\mathbf{x})$ is difficult without also knowing Z_θ, which is typically intractable to compute.

- The score $\nabla \log p_\theta(\mathbf{x}) = -\nabla E_\theta(\mathbf{x})$ does not depend on Z_θ and allows one to sample from p_θ through MCMC methods (e.g. Langevin dynamics).

- If we have $q_{\text{data}}(\mathbf{x})$, we want to train a model $\mathbf{s}_\theta(\mathbf{x})$ to approximate the score.
- Trying to learn the exact score is computationally infeasible for deep neural networks, so instead we learn a slightly perturbed score through *Denoising Score Matching*:

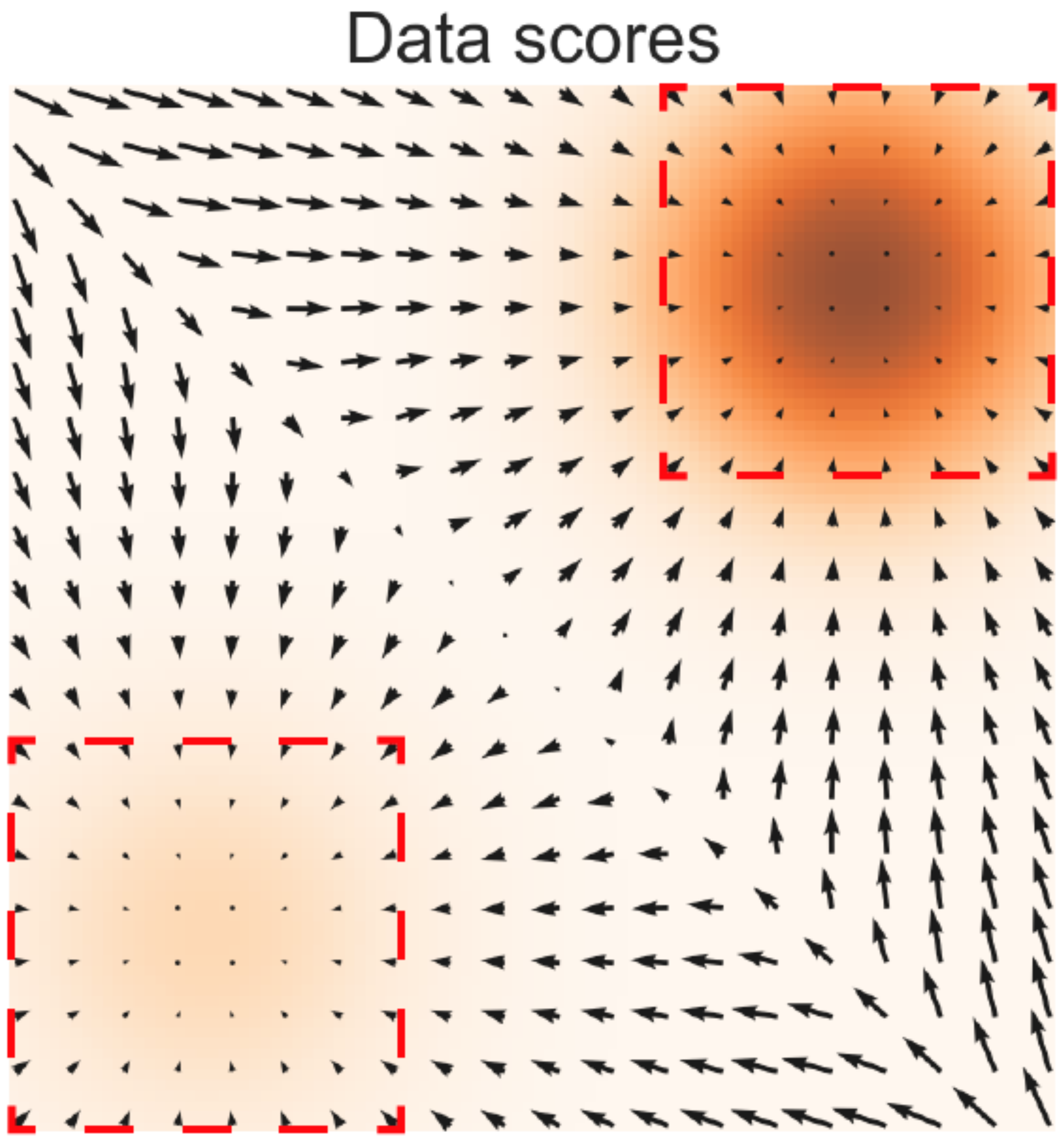
$$\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}),q_{\text{data}}(\mathbf{x})}||\mathbf{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}}\log q_\theta(\tilde{\mathbf{x}} \mid \mathbf{x})||^2$$

where $q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}; \mathbf{x}, \sigma^2\mathbf{I})$. The optimal score network minimizes this when $\mathbf{s}_\theta(\mathbf{x}) = \nabla_{\mathbf{x}}\log q_\sigma(\mathbf{x}) \approx \nabla_{\mathbf{x}}\log q_{\text{data}}(\mathbf{x})$.

- Can sample with Langevin Dynamics:

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \frac{\epsilon}{2}\nabla_{\mathbf{x}}\log q_\sigma(\tilde{\mathbf{x}}_{t-1}) + \sqrt{\epsilon}\mathbf{z}_t$$

- Are we done? No, there are still problems with this method.
- Langevin dynamics can theoretically sample the distribution, but that may take a long time. Slow mixing can occur if modes are separated.



The two modes are separated by low-density regions,
and so Langevin dynamics may not easily sample both.

- For large σ, the distribution q_σ suffers less from these problems and are easier to sample. In fact, for very large σ the distribution looks like a normal distribution with very high variance.

- Solution: create a family of perturbed distributions parameterized by $\sigma_1 < \sigma_2 < \cdots < \sigma_N$ and learn the score for each. Start sampling from the noisiest distribution using Langevin dynamics first and then move to a less noisy distribution.

- At the end the final sample from q_{σ_1} should be an approximate sample of q_{data}.

- The model that learns the scores of this family of distributions is sometimes known as a Noise Conditional Score Network (NCSN).

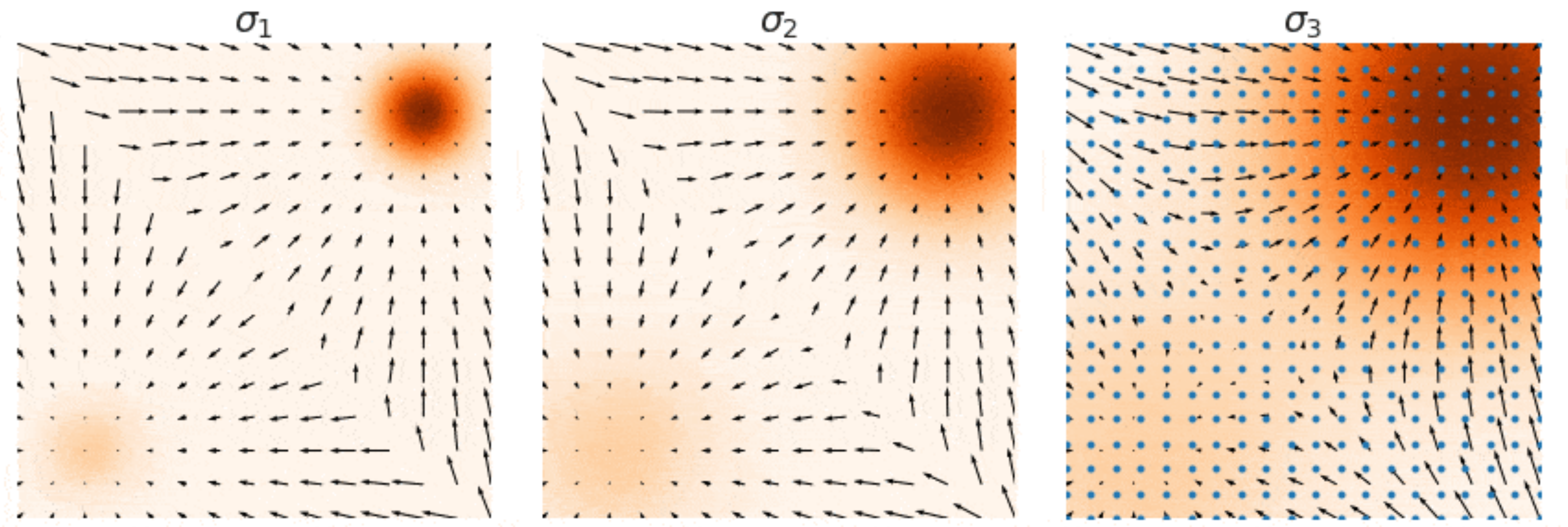
- Since $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = -(\tilde{\mathbf{x}} - \mathbf{x})/\sigma^2$, the denoising score matching objective for a fixed σ is given by

$$\ell(\theta; \sigma) := \mathbb{E}_{q_{\text{data}}(\mathbf{x}), \tilde{\mathbf{x}} \sim \mathcal{N}(\mathbf{x}, \sigma^2 \mathbf{I})} \left\| \mathbf{s}_\theta(\tilde{\mathbf{x}}, \sigma) + \frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2} \right\|^2$$

- A typical choice for the full training objective is a weighting of $\ell(\theta, \sigma)$ based on the expected norm of the score ($\|\mathbf{s}_\theta(\mathbf{x}, \sigma)\| \propto 1/\sigma$)

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sigma_i^2 \ell(\theta; \sigma_i)$$

- Use gradient descent on $L(\theta)$ to train. Sampling is done by Langevin dynamics at each noise level, passing the sample to the lower noise levels.



Sampling from NCSN involves using Langevin dynamics at each level. Samples from higher level get passed as initial points to lower levels.

Generalizing DDPMs and NCSNs

- So far we have seen two ways to mathematically define diffusion models. Both relied on discrete steps in time or noise.
- We can reformulate the noising/denoising into a continuous process. This has a number of advantages:
  - DDPMs and NCSNs become discretizations of this continuous process.
  - More control on the speed/quality of sampling.
  - Simple to formulate controllable generation.
  - There is a deterministic way to sample distribution.

- Differential Equation

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}, t)$$

- Stochastic Differential Equation (SDE)

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}, t) + g(t)\frac{\mathrm{d}\mathbf{W}}{\mathrm{d}t}$$

White Noise = “Derivative of Gaussian Random variable”

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{W}$$

- If $\mathbf{x}(0) \sim p_0$ is the data distribution, then the SDE $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{W}$ will perturb this distribution with white noise.

- Typically, for a long period of time T the distribution of $\mathbf{x}(T) \sim p_T$ will have almost no information about the initial distribution (most often a normal distribution).

- The SDE is thus describing the *forward process* of the diffusion model. It is describing a continuous way of adding noise.

- The backward process will also be given by an SDE with initial condition given by $\mathbf{x}(T) \sim p_T$.

- To sample, we solve a reverse-time SDE

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{W}}$$

which is guaranteed to have the same distributions as the forward SDE.

- Want to learn the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ for $0 \leq t \leq T$. For fixed t the objective becomes

$$\mathbb{E}_{\mathbf{x}(0)}\mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)}||\mathbf{s}_\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0))||^2$$

where $p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0))$ is the density function of the solution given the initial value is fixed at $\mathbf{x}(0)$.

- Most models choose the SDE so that p_{0t} has an exact formula and can be sampled.

- We can derive continuous analogues of both models that we've seen by taking the limit as the step sizes decrease to 0.
- DDPM
    - The parameters β_i become a continuous function $\beta(t)$ and the Markov chain becomes an SDE
    - Variance Preserving SDE: $\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{W}$
- NCSN
    - The noise parameters σ_i become an increasing, continuous function $\sigma(t)$
    - Variance Exploding SDE: $\mathrm{d}\mathrm{x} = \sqrt{\frac{\mathrm{d}[\sigma^2(t)]}{\mathrm{d}t}}\mathrm{d}\mathbf{W}$

Further Reading

- “Generative Modeling by Estimating Gradients of the Data Distribution” by Song, Ermon (see also the blog post of the same name by Song)
- “Score-based Generative Modeling through Stochastic Differential Equation” by Song, et al.