UNIT 6

Flow Matching

Johannes Brandstetter
Institute for Machine Learning

JOHANNES KEPLER
UNIVERSITY LINZ

Institute for
Machine Learning

Copyright statement:

This material, no matter whether in printed or electronic form, may be used for personal and non-commercial educational use only. Any reproduction of this material, no matter whether as a whole or in parts, no matter whether in printed or in electronic form, requires explicit prior acceptance of the authors.

DDPM

- The original DDPM transition probabilities are:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I}\right)$$
$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

- One important observation is that the transition probability $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ follows a Markov chain
- The Markov property ensures that the system is memoryless, i.e., once we know $\boldsymbol{x}_{t-1}$, we know $\boldsymbol{x}_t$
- Downside: a Markov chain can take many steps to converge
- Jonathan Ho et al.

DDIM

- Given:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_t}\boldsymbol{x}_0, (1-\alpha_t)\mathbf{I}\right)$$

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}}\left(\frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}\right), \sigma_t^2\mathbf{I}\right)$$

- We ensure that

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0, (1-\alpha_{t-1})\mathbf{I}\right)$$

- via a family of backward processes:

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t - \alpha_t\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\mathbf{I}\right)$$

- Jiaming Song et al.

DDIM

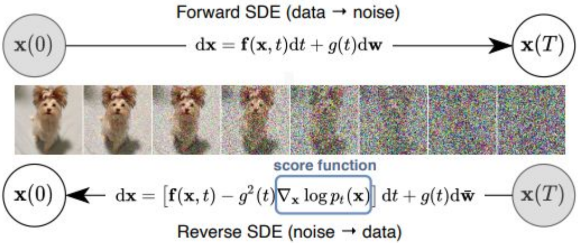
- The forward process can be derived from Bayes' rule:

$$q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0) = \frac{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)} .$$

- Unlike the diffusion process, the forward process is no longer Markovian, since each $\boldsymbol{x}_t$ could depend on both $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_0$
- The magnitude of σ controls the stochasticity of the forward process
- When $\sigma \to 0$, we reach an extreme case where as long as we observe $\boldsymbol{x}_0$ and $\boldsymbol{x}_t$ for some t, then $\boldsymbol{x}_{t-1}$ becomes known and fixed

Score-Based Generative Modeling through Stochastic Differential Equations

- Diffusion models formulated in terms of continuous time Stochastic Differential Equations (SDEs)
- SDE solvers are used to generate samples



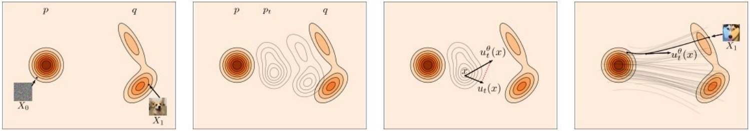
- Yang Song et al.

Flow Matching

- Given access to a training dataset of samples from some target distribution q over $\mathbb{R}^d$:
- Flow Matching (FM) builds a probability path $(p_t)_{0 \leq t \leq 1}$, from a known source distribution $p_0 = p$ to the data target distribution $p_1 = q$, where each p_t is a distribution over $\mathbb{R}^d$
- Flow matching is a simple regression objective to train the velocity field neural network describing the instantaneous velocities of samples along the probability path p_t
- For inference, we generate a novel sample from the target distribution $X_1 \sim q$ by (i) drawing a novel sample from the source distribution $X_0 \sim p$, and (ii) solving the Ordinary Differential Equation (ODE) determined by the velocity field $\boldsymbol{u} : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$

Flow Matching

- The goal of flow matching is to find a flow mapping samples X_0 from a known source or noise distribution p into samples X_1 from an unknown target or data distribution q



(a) Data. **(b)** Path design. **(c)** Training. **(d)** Sampling.

Flow (Mathematics)

- A flow is a function that describes how points in a space move over time according to a set of rules, usually defined by a differential equation or vector field
- Let X be a space $\mathbb{R}^d$, and consider a vector field $\boldsymbol{u} : X \to \mathbb{R}^d$
- The flow $\psi(t, \boldsymbol{x}_0)$ is a function $\psi : \mathbb{R} \times X \to X$, such that:
  - $\psi(0, \boldsymbol{x}_0) = \boldsymbol{x}_0$ (the starting point doesn't move at $t = 0$)
  - $\frac{d}{dt}\psi(t, \boldsymbol{x}_0) = \boldsymbol{u}_t(\psi(t, \boldsymbol{x}_0))$ (the path follows the vector field)
  - $\psi(t + s, \boldsymbol{x}_0) = \psi(t, \psi(s, \boldsymbol{x}_0))$ (consistency over time)
- $\psi(t, \boldsymbol{x}_0)$ = $\psi_t(\boldsymbol{x}_0)$ (both notations are common)

The Flow in Flow Matching

- In Flow Matching, an ODE is defined via a time-dependent vector field $\boldsymbol{u} : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$
- The velocity field $\boldsymbol{u}$ determines a time dependent flow $\psi : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$
- The flow is defined as:

$$\frac{d}{dt}\psi_t(\boldsymbol{x}_0) = \boldsymbol{u}_t(\psi_t(\boldsymbol{x}_0))$$

- with $\psi_t(\boldsymbol{x}_0) = \psi(t, \boldsymbol{x}_0)$ and $\psi_0(\boldsymbol{x}_0) = \boldsymbol{x}_0$
- The velocity field $\boldsymbol{u}_t$ generates the probability path p_t if its flow ψ_t satisfies: $X_t = \psi_t(X_0) \sim p_t$ for $X_0 \sim p_0$
- The velocity field $\boldsymbol{u}_t$ is the only tool necessary to sample from p_t by solving the ODE $\frac{d}{dt}\psi_t(\boldsymbol{x}_0) = \boldsymbol{u}_\theta^t(\psi_t(\boldsymbol{x}_0))$

Designing The Probability Path

- The goal of Flow Matching is to learn the parameters θ of a velocity field $\boldsymbol{u}_\theta$
- We need to: (i) design a a probability path that interpolates between p and q, and (ii) train a velocity field $\boldsymbol{u}_\theta^t$ that generates p_t by means of regression
- Let's assume the source distribution $p = p_0 = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
- We construct the probability path p_t as the aggregation of the conditional probability paths $p_{t|1}(\boldsymbol{x}|\boldsymbol{x}_1)$, each conditioned on one of the data examples $X_1 \sim q = \boldsymbol{x}_1$ comprising the training dataset.

Designing The Probability Path

- The probability path p_t therefore follows the expression:

$$p_t(\boldsymbol{x}) = \int p_{t|1}(\boldsymbol{x}|\boldsymbol{x}_1) q(\boldsymbol{x}_1) d\boldsymbol{x}_1 \ ,$$

- where $p_{t|1}(\boldsymbol{x}|\boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{x}|t\boldsymbol{x}_1, (1-t)^2\boldsymbol{I})$
- This path is known as the **conditional optimal-transport** or linear path
- Several other path choices exist, but we focus on the conditional optimal-transport path for simplicity
- We define the random variable $X_t \sim p_t$ by drawing $X_0 \sim p$, drawing $X_1 \sim q$, and taking their linear combination:

$$X_t = tX_1 + (1-t)X_0 \sim p_t$$

Flow Matching Loss

- Second step in Flow Matching: we regress the velocity field $\boldsymbol{u}_t^\theta$ to a target velocity field $\boldsymbol{u}_t$, which generates the desired probability path p_t
- The Flow Matching loss reads:

$$\mathcal{L}_{\mathsf{FM}} = \mathbb{E}_{t,X_t} \|\boldsymbol{u}_t^\theta(X_t) - \boldsymbol{u}_t(X_t)\|^2 \ ,$$

- where $t \sim \mathcal{U}(0,1)$ and $X_t \sim p_t$
- In practice, one can rarely implement the objective above, because $\boldsymbol{u}_t$ is a complicated object governing the joint transformation between two high-dimensional distributions

Conditional Flow Matching Loss

- The Flow Matching objective simplifies drastically by conditioning the loss on a single target example $X_1 = \boldsymbol{x}_1$ picked at random from the training set
- Using

$$X_{t|1} = t\boldsymbol{x}_1 + (1-t)X_0 \quad \sim \quad p_{t|1}(\cdot|\boldsymbol{x}_1) = \mathcal{N}(\cdot|t\boldsymbol{x}_1, (1-t)^2\boldsymbol{I})$$

$$\frac{d}{dt}X_{t|1} = \boldsymbol{u}_t(X_{t|1}|\boldsymbol{x}_1)$$

- we get

$$\frac{d}{dt}X_{t|1} = \boldsymbol{x}_1 - X_0$$

$$X_0 = \frac{X_{t|1} - t\boldsymbol{x}_1}{1-t}$$

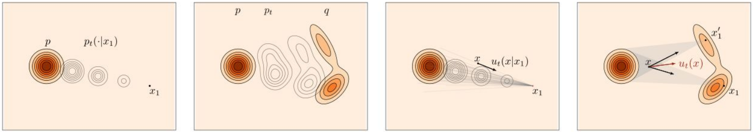
$$\boldsymbol{u}_t(X_{t|1}|\boldsymbol{x}_1) = \boldsymbol{x}_1 - X_0 = \boldsymbol{x}_1 - \frac{X_{t|1} - t\boldsymbol{x}_1}{1-t}$$

Conditional Flow Matching Loss

- Thus, the conditional vector field $\boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{x}_1) = \frac{\boldsymbol{x}_1 - \boldsymbol{x}}{1-t}$ generates the conditional probability path $p_{t|1}(\cdot|\boldsymbol{x}_1)$
- The conditional Flow Matching loss reads:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, X_t, X_1} \left\| \boldsymbol{u}_t^{\theta}(X_t) - \boldsymbol{u}_t(X_t|X_1) \right\|^2 ,$$

- where $t \sim \mathcal{U}(0,1)$, $X_0 \sim p$, $X_1 \sim q$ and $X_t = (1-t)X_0 + tX_1$



(a) Conditional probability path $p_t(x|x_1)$. **(b)** (Marginal) Probability path $p_t(x)$. **(c)** Conditional velocity field $u_t(x|x_1)$. **(d)** (Marginal) Velocity field $u_t(x)$.

Conditional Flow Matching Loss

- Remarkably, the objectives $\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,X_t} \|\boldsymbol{u}_t^\theta(X_t) - \boldsymbol{u}_t(X_t)\|^2$ and $\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t,X_t,X_1} \|\boldsymbol{u}_t^\theta(X_t) - \boldsymbol{u}_t(X_t|X_1)\|^2$ provide the same gradients to learn $\boldsymbol{u}_t^\theta$
- $\nabla_\theta \mathcal{L}_{\text{FM}}(\theta) = \nabla_\theta \mathcal{L}_{\text{CFM}}(\theta)$
- Thus, we can train with the conditional Flow Matching objective, obtain the vector field $\boldsymbol{u} : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ and generate samples by solving the flow $\frac{d}{dt}\psi_t(\boldsymbol{x}) = \boldsymbol{u}_t(\psi_t(\boldsymbol{x}_t))$ via an ODE solver

The bigger picture

- DDPM – based on $\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t)$:
  - Training: $\nabla_\theta \|\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t) - \boldsymbol{x}_0\|^2$
  - Inference:
    $$\boldsymbol{x}_{t-1} = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{(1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t) + \sigma_q(t)\boldsymbol{z}$$
- DDPM – based on $\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t)$:
  - Training: $\nabla_\theta \|\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t) - \boldsymbol{\epsilon}_0\|^2$
  - Inference: $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t) + \sigma_q(t)\boldsymbol{z}$
- DDPM – v-prediction
  - More advanced noise scheduler which uses a linear combination of $\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t)$ and $\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t)$
- Flow Matching
  - Training objective is also a linear combination (for conditional optimal transport) of noise and target

The bigger picture

- Recent works (e.g., Scaling Rectified Flow Transformers for High-Resolution Image Synthesis) argue that all Diffusion and Flow Matching losses and inference schemes can be re-written to some weighted regression on $\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t)$
- Frameworks like Stochastic Interpolants unify Diffusion and Flow Matching algorithms (SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers)