Preferred Networks

神経回路学会2023 全国大会 基調講演

拡散と流れに基づく学習と推論

Learning and Inference Based on Diffusion and Flow

Preferred Networks 岡野原 大輔

@hillbig

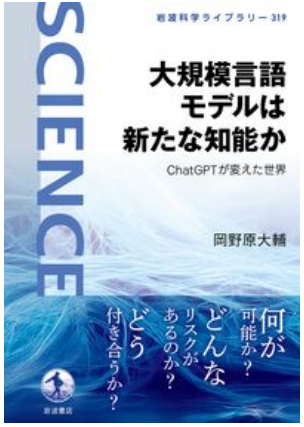
自己紹介：岡野原　大輔

Preferred Networks 代表取締役 最高研究責任者
Preferred Computational Chemistry 代表取締役社長
Preferred Robotics 取締役

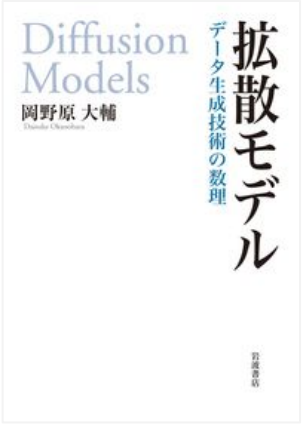
X(twitter): @hillbig

日経RoboticsでAI最前線を連載中（2015年より毎月）

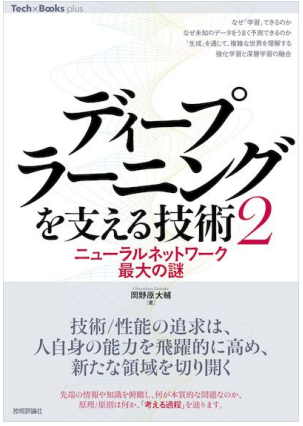
関連書籍



岩波書店 2023
一般向け



岩波書店 2023
専門家向け



技術評論社 2021 2022
ディープラーニングの基礎知識

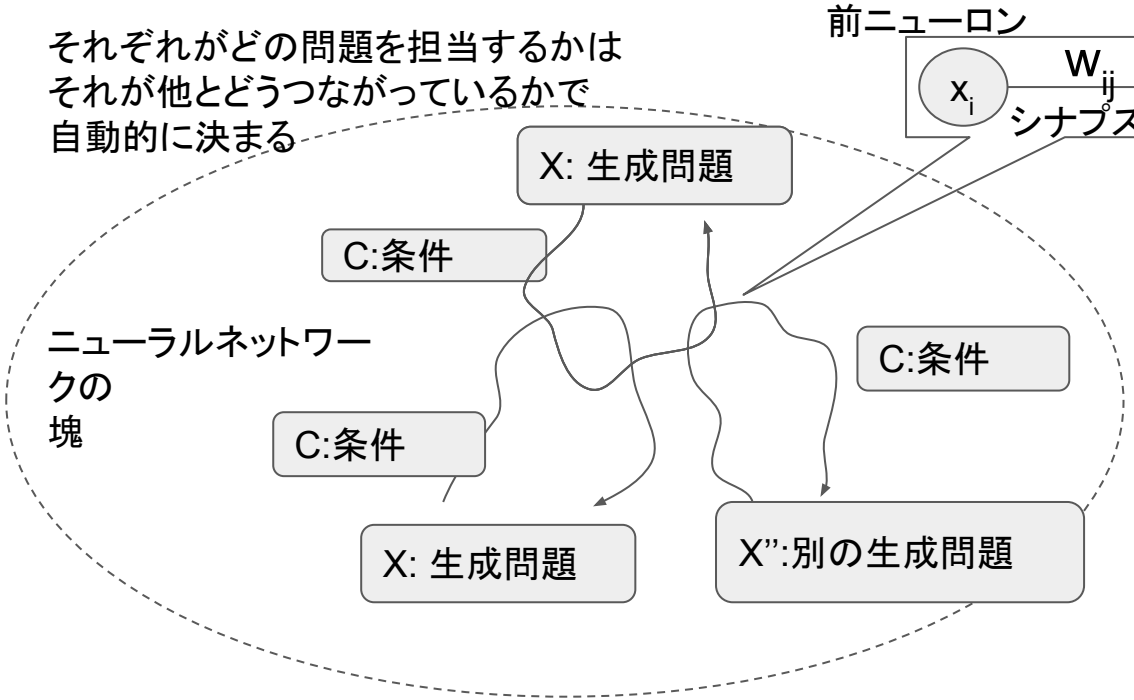
日経BP 2022
個別の深い話題

↑今日の話題はこれが多め

アジェンダ

- 生成モデル
- 拡散モデル / フローマッチング
- エネルギーベースドモデルによる学習
  - 予測符号化、対比ヘブ則、均衡伝播法

今日のゴール：脳内での学習モデルの仮説



$$\Delta w_{ij} \propto [\rho(x_i)\rho(x_j)]$$

様々な条件付生成問題を非同期・同時多発的におきる場合の学習の枠組みを提唱する。
各シナプスは周囲で何が起きているのか（どの学習がどこで進行しているのか）を知らずに、その前後のニューロンの情報だけを使いシナプスを更新する

何を学習するのか：条件付生成
→拡散モデル
どのように学習するのか：
→エネルギーベースモデル

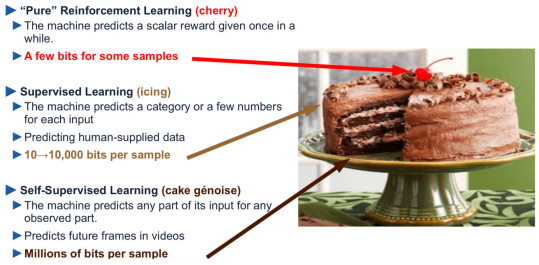
条件付生成モデル

$x \sim p(X \mid C)$

X: 生成対象　C: 条件

- 生成モデル：対象ドメインのデータを生成できるようなモデル
  - テキスト、画像、動画、化合物、行動列 等
- **条件**を通じて、制約、指示、対象ドメインなどを指定する
  - 条件付した方が分布が単純になり学習や推論が簡単になる
- 条件数や生成対象数は可変の場合が多い
- 大規模言語モデル、拡散モデルによる画像/音声生成など

自己教師あり学習



[Lecun 2019]

正解がタダでいくらでも手に入る教師あり学習問題

- 条件付け生成
  - 時系列予測：過去から未来を予測する
  - 欠損補間：一部を欠損させ残りから欠損を予測
- 対比：意味が同じものと違うものを対比させる

そのタスクの達成自体が目標ではなく、そのタスクを達成するための副作用として別の能力を獲得する

教師あり学習と違って、膨大かつ多様なデータを利用でき、特定タスク向けでない理解ができるようになる

例：大規模言語モデル(LLM)の自己教師あり学習

「こうしたことから、私は父と一緒に＊＊＊へ行き相談した」

***に入る単語を予測するには？
私や父はどういう人か、「こうしたこと：とは何か
行って相談する場所はなにか　など

（少しでも）うまく予測するためには内容を理解する必要がある

言語モデルは次の単語を予測するために文章や概念を理解する必要に駆られ、結果として言語やその背後の概念を理解している

例：拡散モデルの自己教師あり学習

（後で詳しく解説）画像のデノイジングという自己教師あり学習

- 補完、超解像、Zero-shot編集が可能
- その他アプリケーションとして密度推定、非可逆圧縮、敵対的摂動頑健性向上、最適化などでも最高精度を達成
- 学習時に使わなかった別情報での条件付生成を少量のデータで適応できる

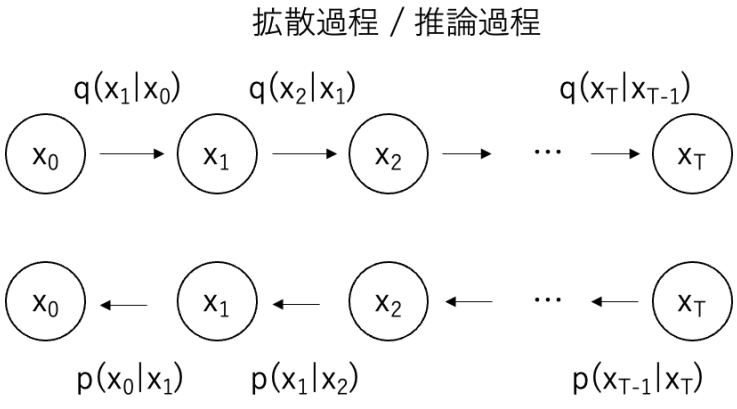
  深度、3D、スタイル

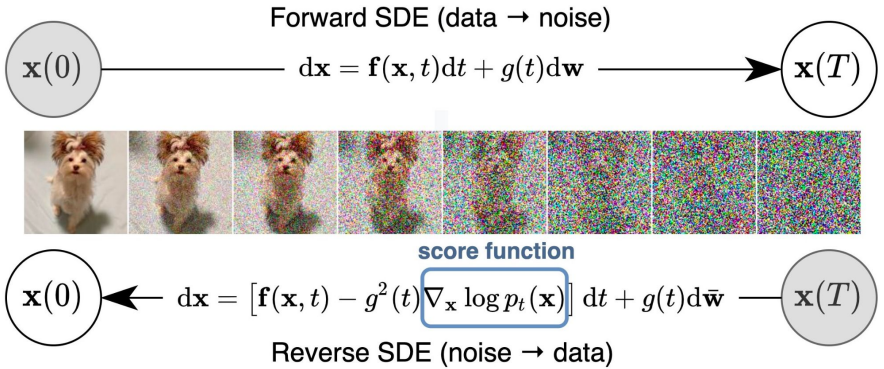
ControlNet [Zhang+ 2023]

拡散モデル / フローマッチング

拡散モデル [SohI-Dickstein+ 2015] [Song+ 2019] [Ho+ 2020]

- 非平衡熱力学を源流に持つ、深層生成モデルの一種
- データにノイズを徐々に加えていく拡散過程を逆向きに辿る逆拡散過程（生成過程）によって生成モデルを定義する
- データを破壊することで生成方法を学習する





https://yang-song.net/blog/2021/score/

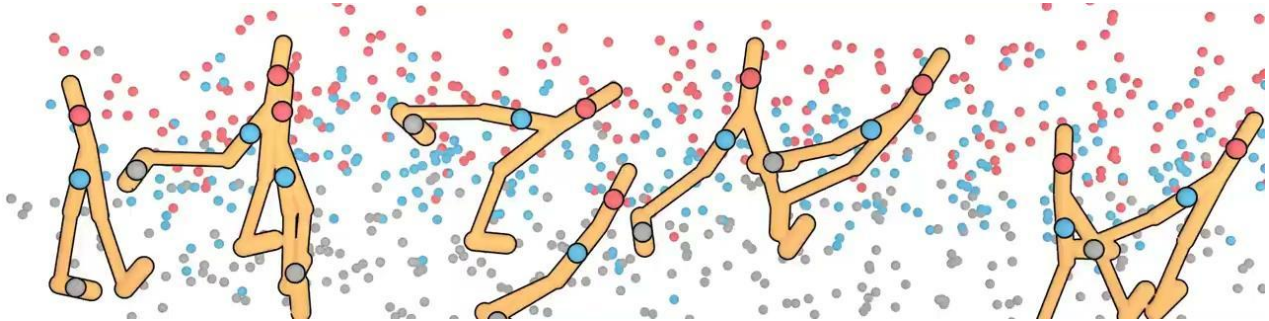
拡散モデルの広がり

- 多くのタスクで従来性能を凌駕する性能を達成
  - 画像、音声、点群、化合物、動画
  - 編集：補間、超解像、Zero-shot編集
  - 密度推定、非可逆圧縮、敵対的摂動頑健性向上、最適化
- テキスト条件付画像生成では既に1億枚超の画像が生成されている

テキスト条件付動画生成の例
[Ho+ 2022]

midjourney v5の出力例 Yuki Homma @y__homm https://twitter.com/y__homm/status/1636186478899494912

Diffuser 拡散モデルによる制御の生成 [Janner+2022]



height( ) > height( )

height( ) > height( )

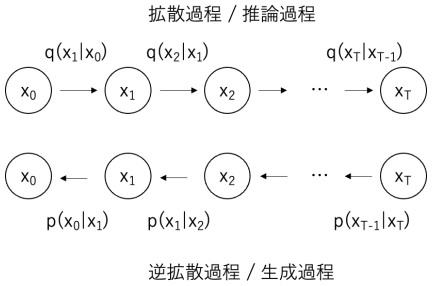
height( ) > height( )

拡散モデルは複数の確率層からなるVAE

- 拡散過程（固定の推論）
- 逆拡散過程（生成）
- データの対数尤度の変分下限（ELBO）最大化で学習

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$



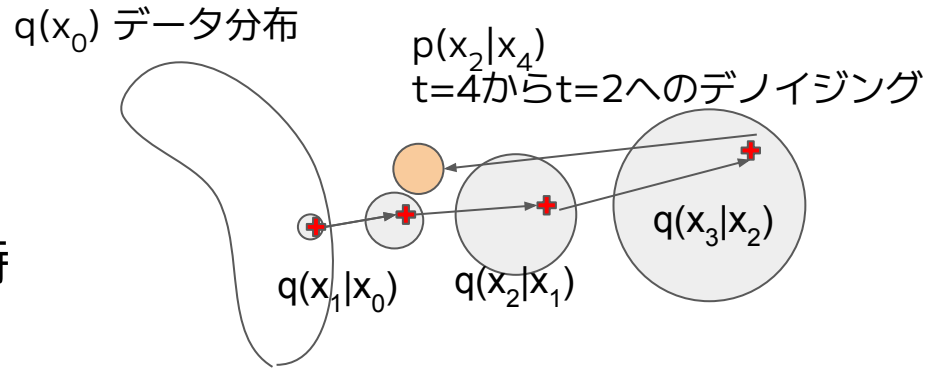
$$\log p_\theta(\mathbf{x}_0) = \log \int p(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}$$

$$\geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log p(\mathbf{x}_T) + \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] := L(\theta)$$

データを拡散して破壊した時に、元に復元できる経路を求める

物理の言葉でいえば、事前分布から目標分布へ変換する経路の中で発生する散逸（自由エネルギー減少）が最小の経路を求める

デノイジング



拡散過程が次のように与えられた時

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\alpha_t \mathbf{x}_0, \sigma_t^2 \mathbf{I})$$

最適な逆拡散過程の1ステップ(s<t)は次のように求められる

$$p(\mathbf{x}_s|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_s; \mu(\mathbf{x}_t, s, t), \tilde{\sigma}^2(s,t)\mathbf{I})$$

$$\mu(\mathbf{x}_t, s, t)$$

$$= \frac{\alpha_{t|s}\sigma_s^2}{\sigma_t^2}\mathbf{x}_t + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2}\hat{\mathbf{x}}_\theta(\mathbf{x}_t; t)$$

今のデータに推定した**デノイジング結果**を足す

または

$$= \frac{1}{\alpha_{t|s}}\mathbf{x}_t - \frac{\sigma_{t|s}^2}{\alpha_{t|s}\sigma_t}\hat{\epsilon}_\theta(\mathbf{x}_t; t)$$

今のデータから推定した**ノイズ**を引く

または

$$= \frac{1}{\alpha_{t|s}}\mathbf{x}_t + \frac{\sigma_{t|s}^2}{\alpha_{t|s}}\mathbf{s}_\theta(\mathbf{x}_t; t)$$

今のデータを推定した**スコア**に従って遷移する

導出は拡散モデル本などを参考

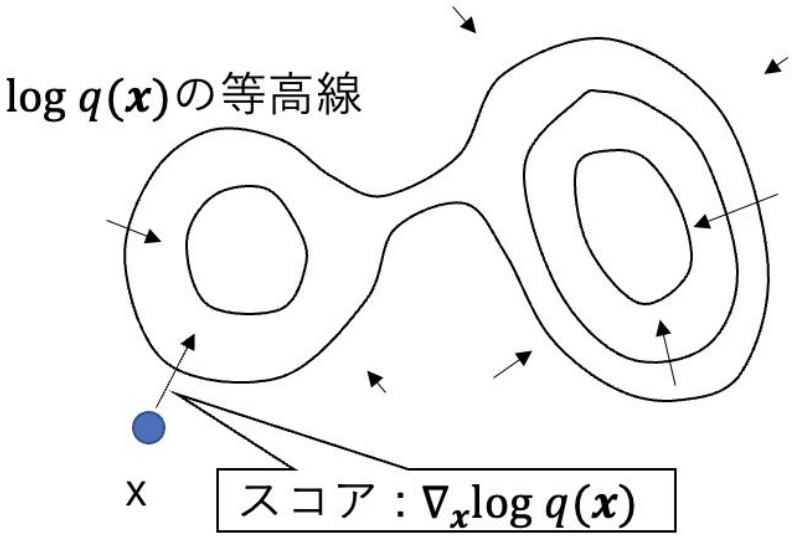
スコア* = 対数尤度の勾配 = エネルギーの負の勾配

*注意：情報幾何などの文脈ではパラメータについての勾配だがここでは入力についての勾配
この名前を付けたのは[Hyvarinen 2005]

$$s(\mathbf{x}) := \nabla_{\mathbf{x}} \log p(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^d$$

$$q_\theta(\mathbf{x}) = \exp(-f_\theta(\mathbf{x}))/Z(\theta)$$

$$\begin{aligned}
&\nabla_{\mathbf{x}} \log q_\theta(\mathbf{x}) \\
&= -\nabla_{\mathbf{x}} f_\theta(\mathbf{x}) - \underbrace{\nabla_{\mathbf{x}} \log Z(\theta)}_{=0} \\
&= -\nabla_{\mathbf{x}} f_\theta(\mathbf{x})
\end{aligned}$$

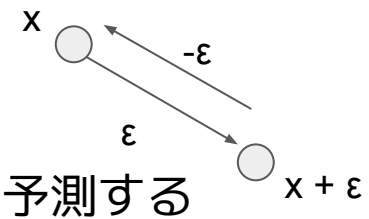


スコアには分配関数が現れず、局所的な情報だけで決定される
→ モデル化（学習）しやすい、合成しやすい、対称性を導入しやすい

デノイジングスコアマッチング [Vincent+ 2011]

以降、簡略化のため拡散過程の時刻tの代わりに撹乱強度σを使う

$$p_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}; \mathbf{x}, \sigma^2\mathbf{I})$$



デノイジングスコアマッチング目的関数を考える

$$J_{DSM_{p_\sigma}}(\theta) = \frac{1}{2}\mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I}),\mathbf{x}\sim p(\mathbf{x})}\left[\left\| -\frac{1}{\sigma^2}\epsilon - s_\theta(\mathbf{x}+\epsilon,\sigma)\right\|^2\right]$$

(3) 加えたノイズの除去を予測する

(1) ノイズを用意

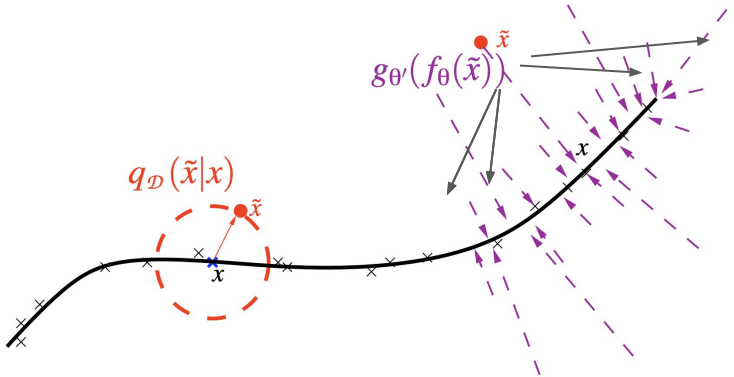
(2) データにノイズを載せる

この最適化問題の最適解$s^*_\theta(x, \sigma)$はスコア$\nabla \log p_\sigma(x)$と一致する

デノイジングスコアマッチングの直感的な意味

ノイズを加えると確率が低い領域へ飛び出す。様々な方向へのデノイジングの平均は確率が高い方向への垂線となる

フォッカー・プランク方程式中にスコアがでてくるのと仕組みは同じであり、エントロピーの変化率からスコアがでてくる



いろいろ方向へ向けてのデノイジングは打ち消し合い垂直方向のみが残る

図は[Vincent 2010]

拡散モデルを使った学習と推論（生成）

学習

データに様々な強度のノイズをのせ、デノイジングできるよう学習する

推論（生成）

完全なノイズからデータをサンプリングし、それをデノイジング強度を下げながらデノイジングするのを繰り返す

局所解に陥らないよう各強度の撹乱後分布でランジュバンモンテカルロを使ってサンプリングするスコアベースドモデル [Song+ 2019]と拡散モデルは目的関数の係数などを除いて一致する [Ho+ 2020]

確率微分方程式での定式化 [Song+ 2021]

ノイズを加える過程のステップ数を無限化した場合の拡散過程は
次の確率微分方程式（ランジュバン方程式）で表される

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

ブラウン運動

この逆拡散過程の確率微分方程式は次の通り与えられる [Anderson 1982]

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

各時刻のスコアさえ求まれば、この方程式に従ってデータを生成できる

前向き確率微分方程式

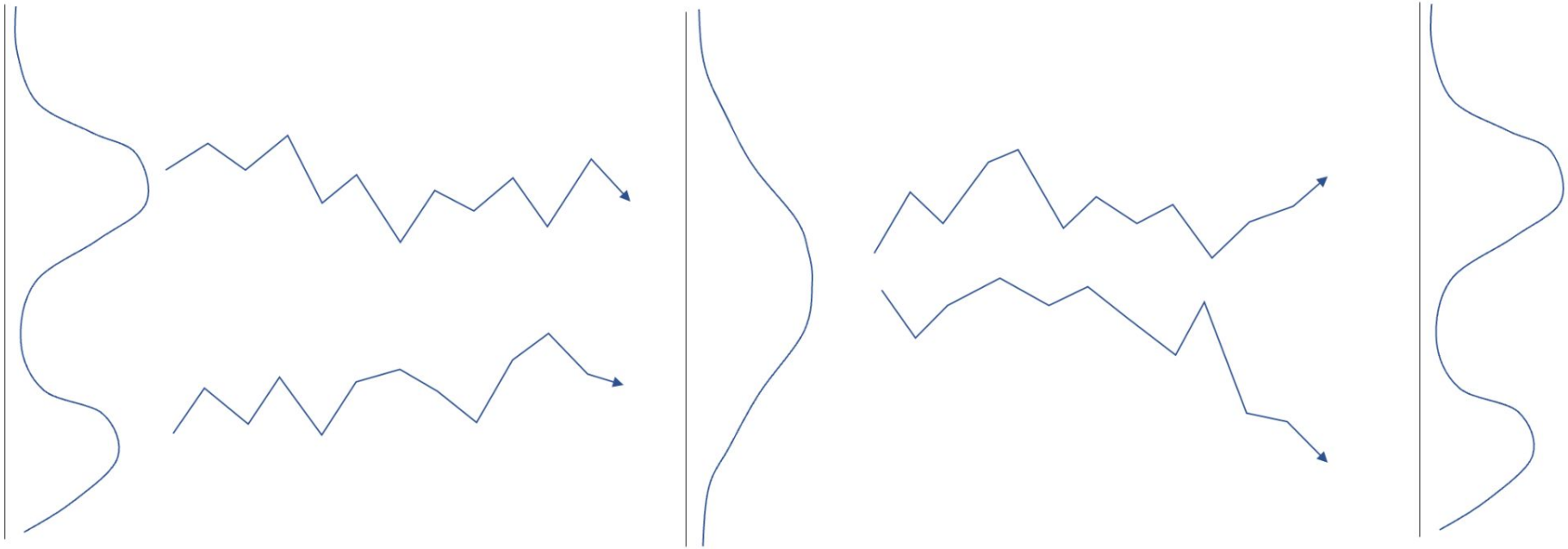
逆向き確率微分方程式

[SDE]

x(0) → x(T) → x(0)

$$dx = f(x,t)dt + g(t)dw$$

$$dx = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]\,dt + g(t)d\bar{w}$$



前向き確率微分方程式でデータ分布は破壊され事前分布に変換され、
逆向き確率微分方程式で、事前分布からデータ分布に変換する。

確率フローODE：常微分方程式での定式化

先程と同じ確率分布を与えるノイズを含まない常微分方程式（Neural ODE）は次のように与えられる [Song+ 2021]

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - \tfrac{1}{2} g^2(t) \nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t$$

事前分布とデータ分布間のデータの1対1対応を与える
実用上は次の拡散係数σ(t)のみに依存する式がよく使われる[Karras+ 2022]

$$\mathrm{d}\mathbf{x} = -\sigma'(t)\sigma(t) \nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma(t))\mathrm{d}t$$

前向き常微分方程式

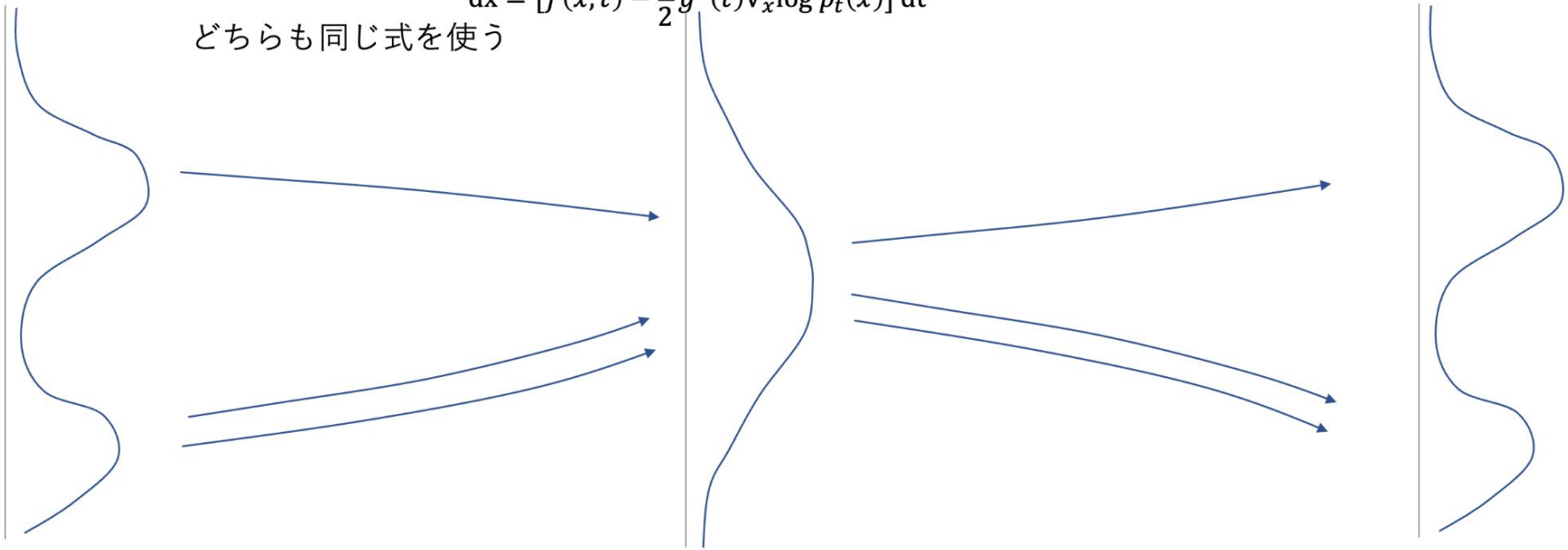
逆向き常微分方程式

[ODE]

x(0) → x(T) → x(0)

$$\mathrm{dx} = [f(x,t) - \frac{1}{2} g^2(t) \nabla_x \log p_t(x)]\,\mathrm{dt}$$

どちらも同じ式を使う



確率微分方程式は常微分方程式で表すことができ、データから事前分布中の点への変換は可逆変換で表すことができる。

なぜ拡散モデルが優れているか (1/5)
一つの最適化問題で安定して学習できる

- VAE ：生成モデルと認識モデルを同時に学習する必要
  - 認識モデルの学習が一般に難しい

- GAN：二つのネットワークを競合して学習する
  - うまく競合させる必要があり、失敗することも多い
  - 逆KL最適化でモード崩壊が起きやすい

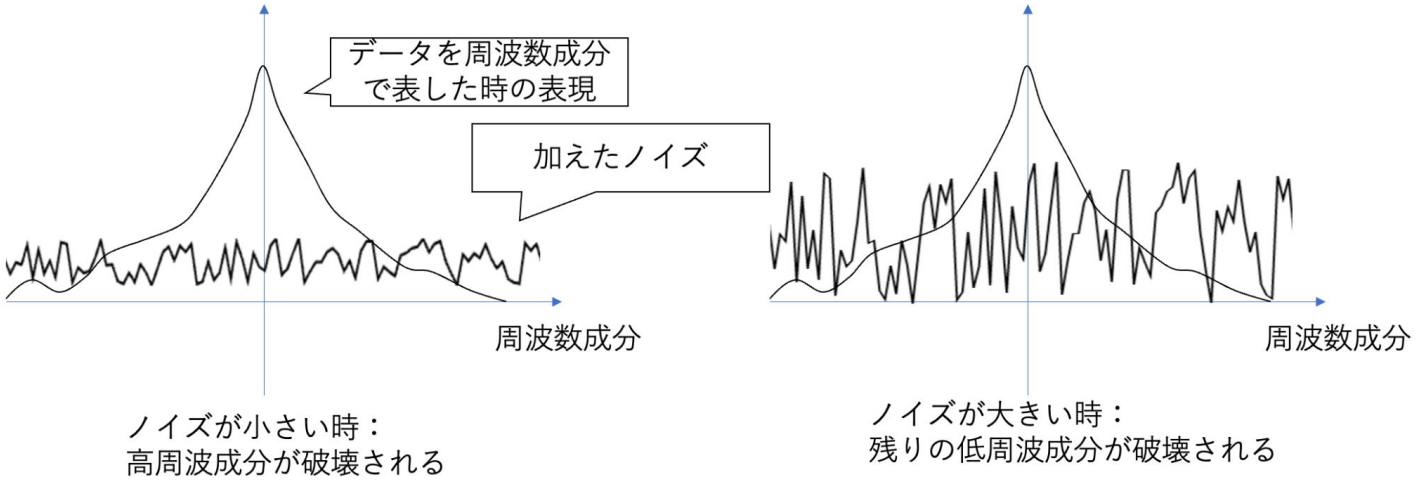
拡散モデルは固定で事後分布崩壊が起きない認識モデル（拡散過程）を使って生成モデルのみ学習するVAE

安定的に学習できる → 大きなデータで長時間学習できる

なぜ拡散モデルが優れているか (2/5)

複雑な生成過程を簡単な部分生成問題に自動的に分解する

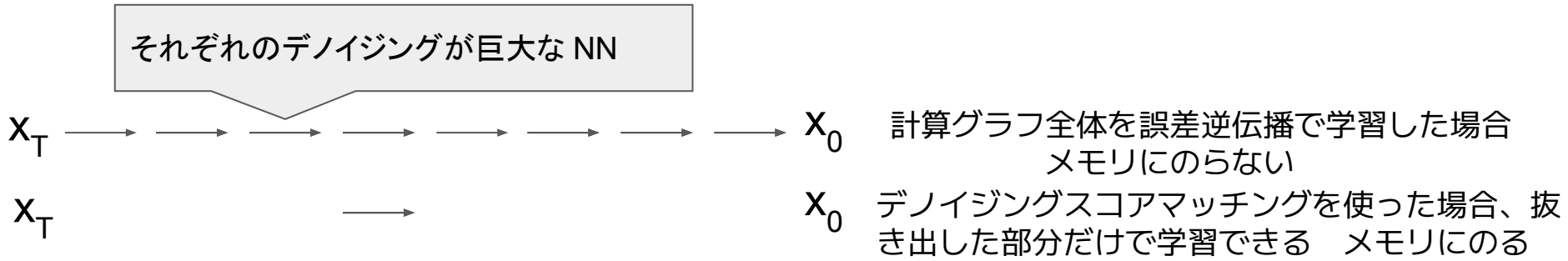
- 拡散過程はノイズを徐々に強くしていくため、データの詳細な部分から全体に向けて徐々に破壊する
- 生成は逆に全体構造から詳細に向けて生成する



なぜ拡散モデルが優れているか (3/5)

非常に長い生成過程を各ステップ独立に学習できる

- 拡散モデルの生成は多くの確率層を経た計算グラフで表される
  - 生成過程がNステップの拡散モデルはN層の確率層からなる
  - 各デノイジングは巨大なNNを使って実現する
- 全過程をまとめて学習することはメモリ容量的に不可能
- デノイジングスコアマッチングは、計算グラフの途中の一部分を抜き出し、そこだけ学習できる。巨大な計算過程を学習可能に
  - このようなアプローチを Simulation Freeとよぶ

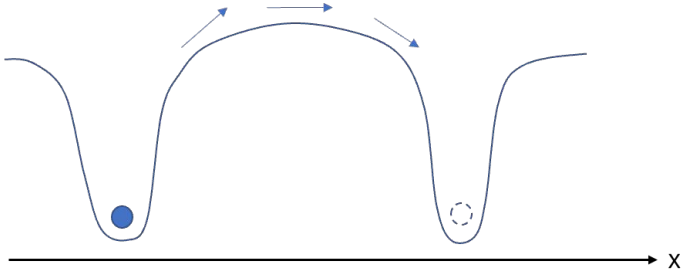


なぜ拡散モデルが優れているか (4/5)
摂動前後のデータの局所的な変化をモデル化

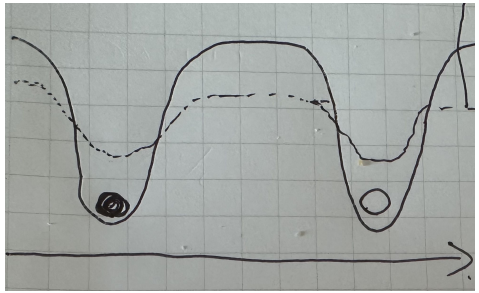
- 分配関数を直接扱う必要がなく、入力空間から入力空間へ変換できる関数だけモデル化すれば良い
  - 他全体は気にせず局所だけモデル化すればよい
- 逆拡散は拡散と同じ関数形で近似できる [Feller 1949]

- 生成時の入力操作に対する対称性を簡単に導入できる
  - 事前分布が操作に対し不変、変換が操作に対し同変な関数を用いた場合、生成確率は操作に対し不変
  - 例：自己回転、並進移動（SE(3)）操作に対し生成確率は不変なモデル [Xu+ 2022]

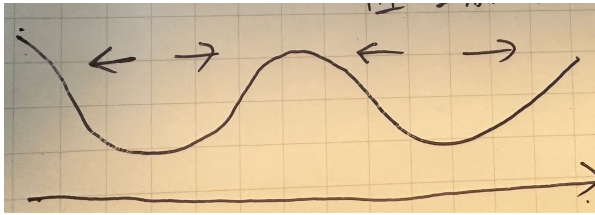
なぜ拡散モデルが優れているか (5/5)
多峰性がある分布からのサンプリングを効率的に生成できる



現在のエネルギーが低い領域から別のエネルギーが低い領域にMCMCで
到達するのは困難



従来の焼きなまし
各点の確率を上下する
（低い位置がどこかは
わからない）



拡散モデルは周囲に広がり
低い場所がどこにあるかを
伝える

拡散過程は高次元空間でどの方向に進めば低い位置に到達できるかを
伝える仕組みがあり、多峰性がある分布から多様性のある推論を可能とする

拡散モデルの条件付生成 [Ho+ 2022]

p(**x** | c) の学習：

$$\left\| -\frac{1}{\sigma^2}\epsilon - s_\theta(\mathbf{x}+\epsilon, c, \sigma) \right\|^2$$

- 条件cを入れたデノイジングスコアマッチングで推定
  - 条件無しもcに特別な値（0ベクトル等）で学習
- 生成時は条件付スコアを使って生成
- 複数の条件付確率が各条件付確率の積で表される場合、そのスコアは和の形で分解され、学習しやすい

$$p(\mathbf{x}|c_1, c_2) \propto p(\mathbf{x}|c_1)p(x|c_2)$$

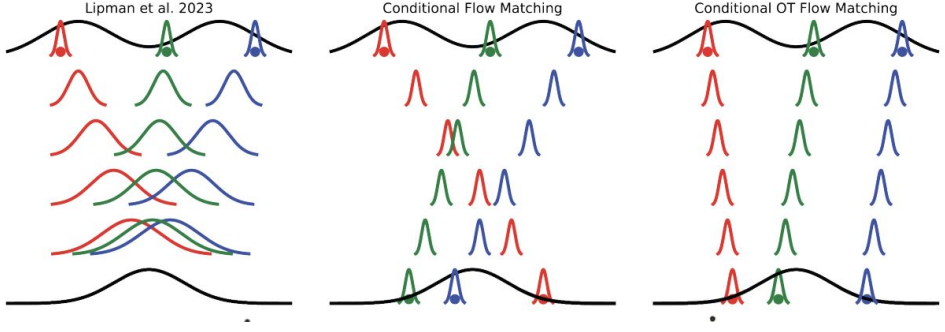
$$s(\mathbf{x}|c_1, c_2) = s(\mathbf{x}|c_1) + s(\mathbf{x}|c_2)$$

テキスト条件づけなど複雑な条件付学習も内部はこのような分解がおきていると考えられる

フローマッチング [Lipman+ 2022] [Tong+ 2023]

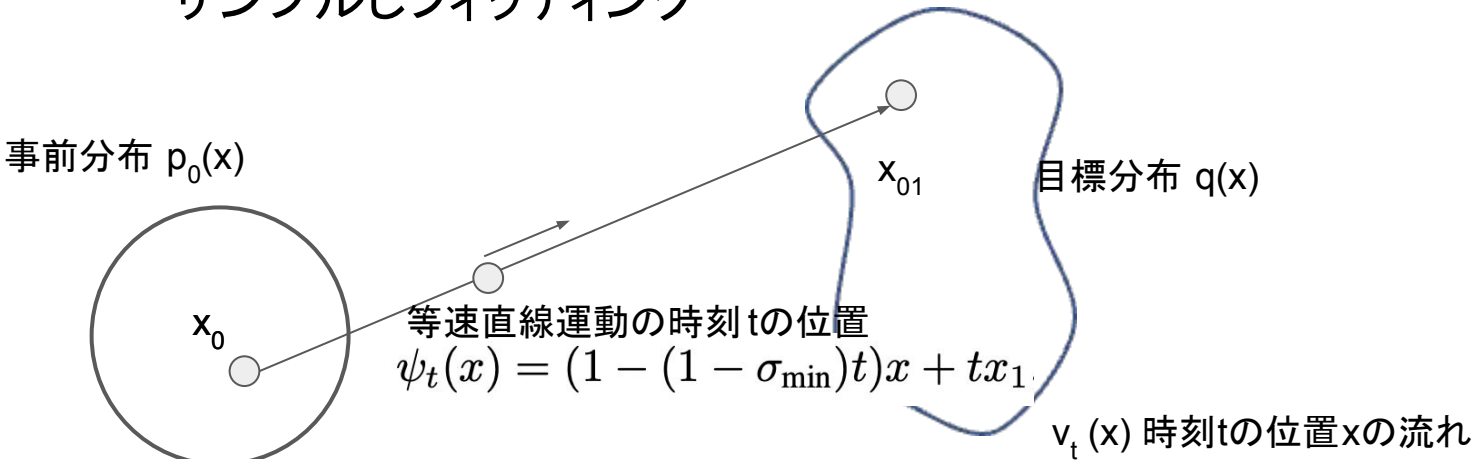
- データ毎のデータ点から事前分布への最適輸送をデータ分布で”周辺化”することでデータ分布から事前分布への最適輸送を求める
  デノイジングスコアマッチングとほぼ同様の仕組みで学習する
- 拡散モデルと同じ枠組みにのるが、フローマッチングが実用上、拡散モデルより優れている部分も多い

拡散項がないため確率微分方程式由来の難しさがない。例えば幾何を導入するのが容易 Riemmanian Flow Matching [Chen+2023]



フローマッチングの学習（最適輸送条件付きフロー版）

1) 事前分布p_0~N(0, σ_{min})からサンプル x_0
2) 目標分布qからサンプル x_T
3) x_0からx_Tまで等速度直線運動するフローの適当な位置をサンプルしフィッティング

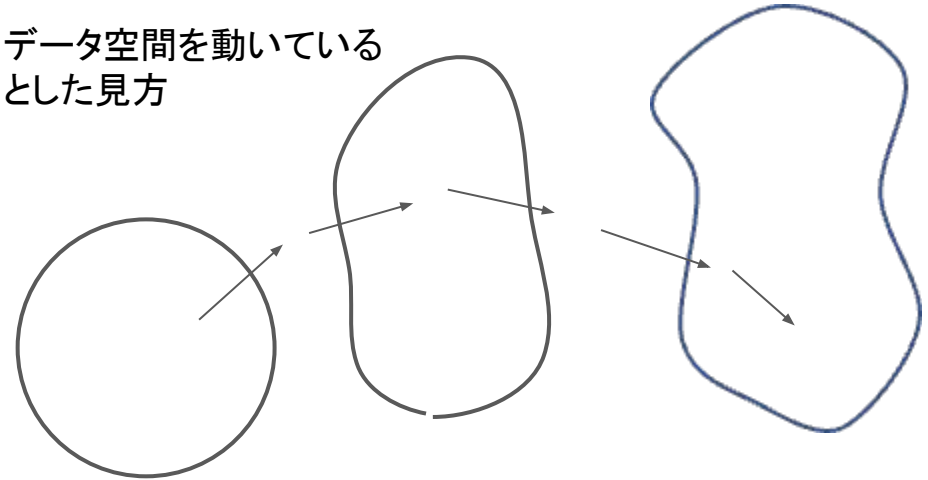


$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)} \left\| v_t(\psi_t(x_0)) - \Big(x_1 - (1 - \sigma_{\min})x_0 \Big) \right\|^2.$$

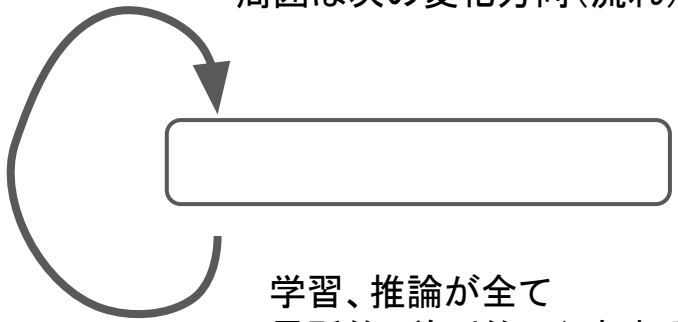
拡散モデル/フローマッチング まとめ

目標分布q(x)を直接モデル化するのではなく、
事前分布$p_0(x)$からのサンプルが目標分布からのサンプルへ
徐々に変化する「流れ」で生成モデルを表す

データ空間を動いている
とした見方

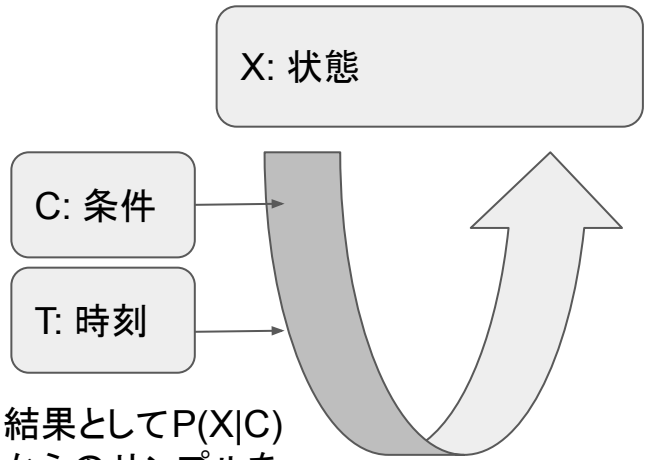


実際の計算ではRNNのように
同じユニットの状態が徐々に変わる
周囲は次の変化方向（流れ）を予測



学習、推論が全て
局所的、並列的に行われる

拡散モデル（フローマッチング）の計算はRNNのようにみなすことができる

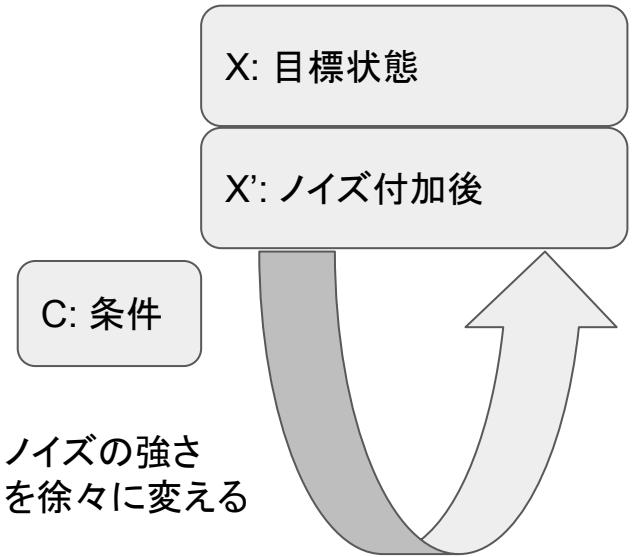


結果としてP(X|C)からのサンプルを得られる

与えられた状態を、条件、時刻に基づいた遷移（流れ）によって徐々に更新していき
P(X|C)からのサンプルを得ることができる

条件は任意に変えることもできる。また条件側も逆に状態とみなし、P(C|X)のように両側の推論はできる

拡散モデル（フローマッチング）の学習シグナルも局所的に得られる



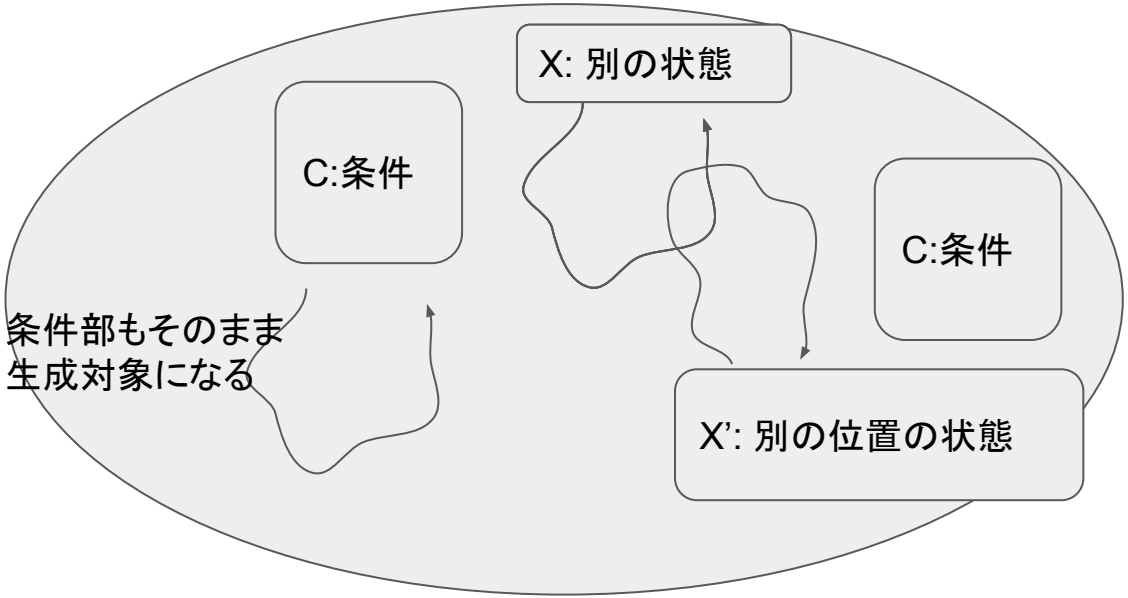
結果としてP(X|C)
からのサンプルを
得られる

ノイズ付加後の状態X'から目標状態Xを推定する
デノイジングによって学習される

例えば状態は連続活性値、ノイズ付加後の状態は
離散化（スパイク）後とみなすことができる

学習誤差を計算する部分は各ニューロン内で
閉じて計算できる（周囲からの予測、自分自身の状態との差）

仮説：脳内でも同時多発的に拡散モデルと同じ仕組みで条件付生成がされているのでは



前述のように条件づけ生成は
予測、推論、分類など多数のタスクをこなすことができるだけでなく、それによって副次的に様々なスキルを自己教師あり学習できる

課題：デノイジングの部分
( )はどのように脳内で
実現可能な形で学習されるのか？
同時多発的に並列にそれぞれの学習ができるようにするには

誤差逆伝播法に変わる学習法

誤差逆伝播法（BP）の脳内実現可能性の問題

BPはNNの学習手法として広く使われているが脳内実現可能性に大きな問題が二つある

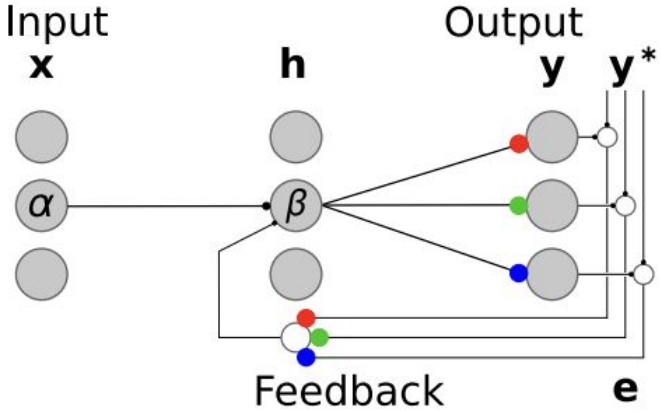
・誤差を正確に逆向きに流す
・学習時と推論時フェースを切り替える

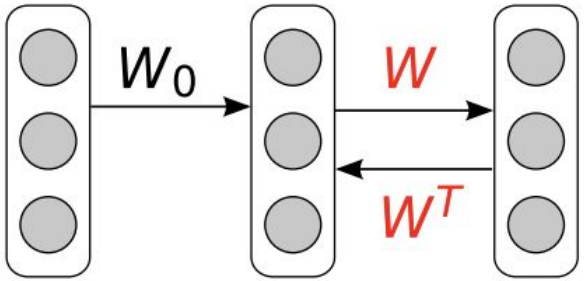
これらは工学的にも問題となっている

BPの問題点 1

誤差を逆向きに流す必要がある

- Weight Transpose Problem
  - BP時に前向き計算時の重み行列の正確な転置を使う必要がある
- 重みも学習しながら変わる中で転置を正確に表すのは困難
- Random Feedbackなど解決案もあるが、スケールした場合に成功していない



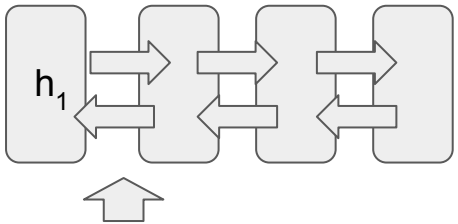


図は[Lillicrap+ 2016]

BPの問題点2

BPは学習と推論を分ける必要がある

- BPは学習フェーズと推論フェーズが分かれており、学習フェーズでは前向き計算後に活性値を覚えておく必要がある
  - 一方、人は学習フェーズだから一定時間考えられないといった制限はない（夜に思い出して学習している仮説などはある）
- 工学的にも活性値を保存する部分、誤差が伝播してくるまで各層の更新ができない部分にスケーラビリティの問題がある



この更新を行うために後続の計算を待つ必要がある [Gomez+ 2022]、かつ活性値h_1を覚えておく必要がある

エネルギーベースモデルによる学習

- エネルギーベースモデルはBPの問題を解決できる

→ 全体的な同期は必要なく局所的な情報のみ使って
非同期・並列に更新できる

→学習フェーズと推論フェーズを分ける必要はない

注：さきほどの拡散モデルにおける
エネルギーとここのエネルギー（デノイジング
をモデル化しようとしている）
違うことを指している

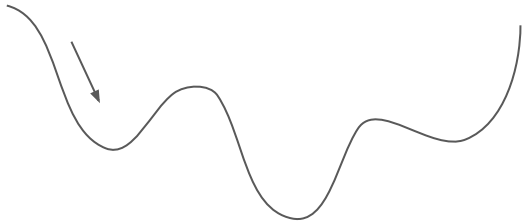
エネルギーベースモデル

エネルギー関数 $E(x_0 \dots x_L, W_0 \dots W_L, T)$
状態 $x_0,\dots,x_L$, パラメータ $W_0,\dots,W_L$

推論 $$\frac{dx_l}{dt} = -\frac{\partial E}{\partial x_l}$$

学習 $$\frac{dW_l}{dt} = -\frac{\partial E}{\partial W_l}$$

エネルギー関数は適当に決めるのでなく、どのような計算したい、どのようなタスクを解きたいことを込めて設計する



エネルギー関数の設計例

$$E = \mathcal{I} + \lambda \mathcal{L}$$

全体エネルギーE は
内部エネルギーと教師あり損失の和
　λは係数　λ=0の場合は教師あり損失無視

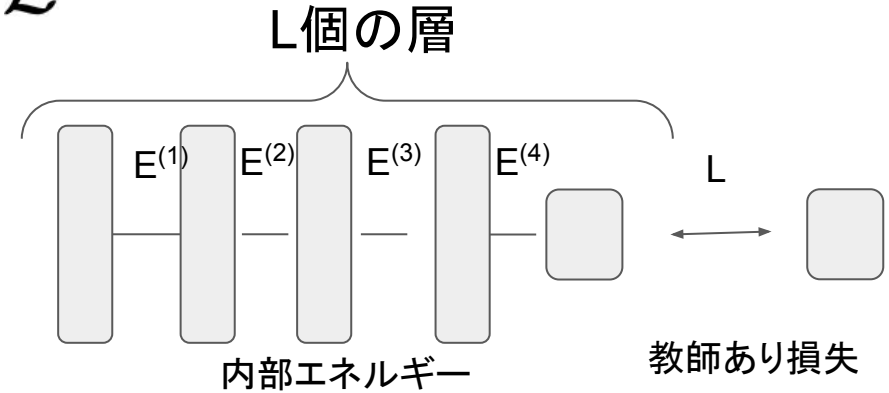
$$\mathcal{I} = \sum_{l=1}^{L-1} E^{(l)}$$

内部エネルギー:（ダイナミクスを決める）
L個の層からなるネットワーク間で定義

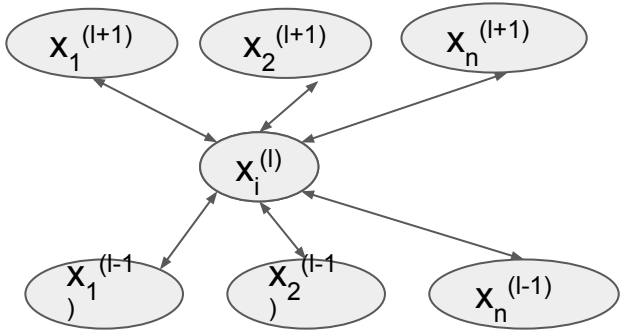
$\mathcal{L}$

教師あり損失:（例:予測　と$\tilde{y}$=解　のy=乗誤差）

$$\|\tilde{y} - y\|^2$$



エネルギーの種類 (½)
ホップフィールド



重みが対称（伝播は両方向に同じ重みである）

$$E^{(l)} = \sum_{ij} w_{ij}^{(l)} \rho(x_i^{(l)}) \rho(x_j^{(l+1)}) - \sum_i b_i^{(l)} \rho(x_i^{(l)}) - \frac{1}{2} \sum_i (x_i^{(l)})^2$$

$$-\frac{\partial E}{\partial x_i^{(l)}} = \rho'(x_i^{(l)}) \left(\underbrace{\sum_j w_{ij}^{(l)} x_j^{(l+1)}}_{\text{上の層からの伝播}} + \underbrace{\sum_j w_{ji}^{(l-1)} x_j^{(l-1)}}_{\text{下の層からの伝播}} - b_i^{(l)} \right) - x_i^{(l)}$$

エネルギーの種類 (2/2)
トランスフォーマーベースドエネルギー

Transformerと似た推論が導出されるエネルギー

$$\mathrm{lse}(\beta, \mathbf{x}) = \frac{1}{\beta} \log \left(\sum_{i=1}^{N} \exp(\beta x_i) \right)$$

$$\frac{\partial \mathrm{lse}(\beta, \mathbf{x})}{\partial x_i} = \mathrm{softmax}(\beta \mathbf{x})_i$$

これより

$$E_{tf} = \mathrm{lse}(\beta, \mathbf{K}^\mathsf{T} \mathbf{x})$$

$$\frac{\partial E_{tf}}{\partial \mathbf{x}} = \mathbf{K} \mathrm{softmax}(\beta \mathbf{K}^\mathsf{T} \mathbf{x})$$

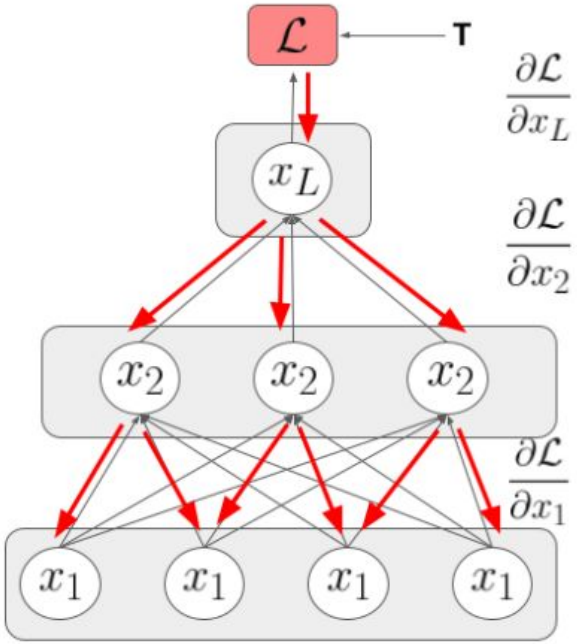
[Ramsauer 2020]

通常のTransformer

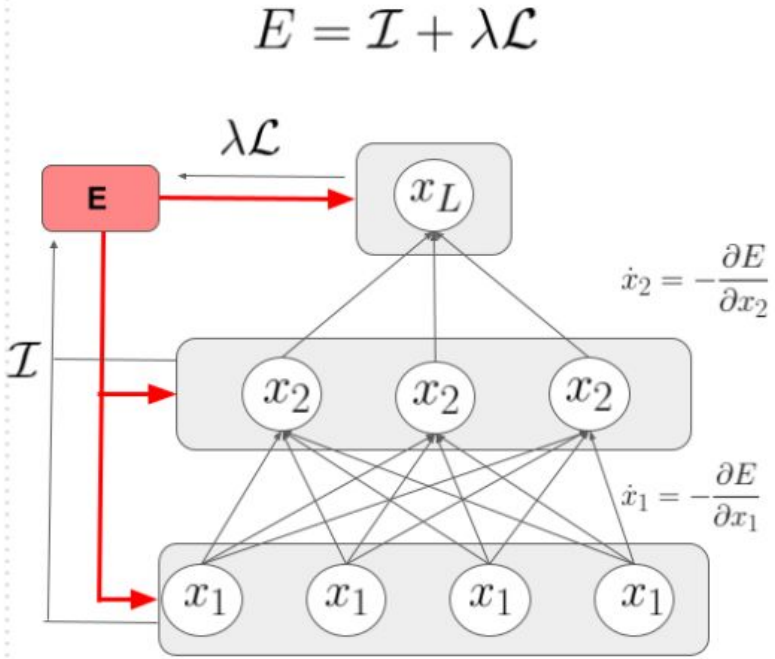
$$\mathbf{V} \mathrm{softmax}(\beta \mathbf{K}^\mathsf{T} \mathbf{x})$$

（通常のTransformerはクエリとキーの近さを元に値を読み込むのに対し、E_{tf}はクエリとキーの近さでキーを読み込む

BPの場合



EPの場合



それぞれの変数の
変化はエネルギーの
勾配に従う

図は[Millidge+ 2023]より

自由相と固有相との差で誤差（教師損失の勾配）が求められる[Millidge 2023]

$$E = \mathcal{I} + \lambda \mathcal{L}$$

λ=0 自由相 教師あり損失を無視し内部エネルギーが収束した状態

λ>0 固定相 教師あり損失を考慮し、予測を教師に「固定」した状態

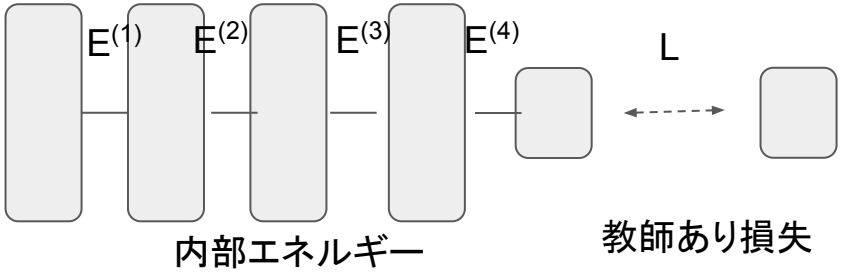
$$\left.\frac{dx}{dt}\right|_{\text{free phase}} = 0 \implies \frac{\partial \mathcal{I}}{\partial x} = 0$$

自由相で収束した状態では内部エネルギーの勾配は0

$$\left.\frac{dx}{dt}\right|_{\text{clamped phase}} = -\frac{\partial E}{\partial x} = -\frac{\partial \mathcal{I}}{\partial x} - \lambda \frac{\partial \mathcal{L}}{\partial x}$$

$$\frac{\partial \mathcal{I}}{\partial x} = 0 \implies \frac{dx}{dt} = -\lambda \frac{\partial \mathcal{L}}{\partial x}$$

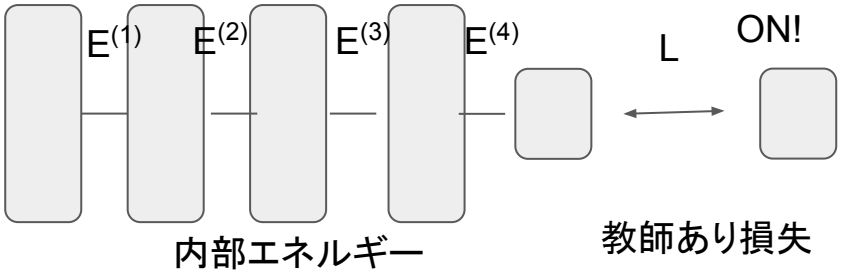
自由相で収束した状態から
固定相に変化した瞬間の
状態の変化量は損失の勾配
（BPにおける誤差）と一致



自由相：教師あり損失由来のエネルギーを無視した状態で収束させる

$$\frac{dx}{dt} = -\lambda \frac{\partial \mathcal{L}}{\partial x}$$

誤差逆伝播法で流れてくる誤差そのもの



自由相から固定相に変化する瞬間
収束していた各状態x_iは次の収束する状態に向けて一斉に変化し始める。この瞬間の変化量が
Lの勾配と一致する
First step algorithm [Song 2020]

自由相と固有相の収束点

同様に、自由相と固定相の収束点が近ければ、固定相で求めた勾配と自由相で求めた勾配の差で教師あり損失の勾配が求まる（並列、非同期に求まる）

過去の様々な学習方法が統一的に説明できる [Millidge+ 2023]

- 対比ヘブ則 [Xie & Seung 2003]
- 弱フィードバックあり予測符号化 [Whittington 2017]
- 均衡伝播法 [Scellier 2017]

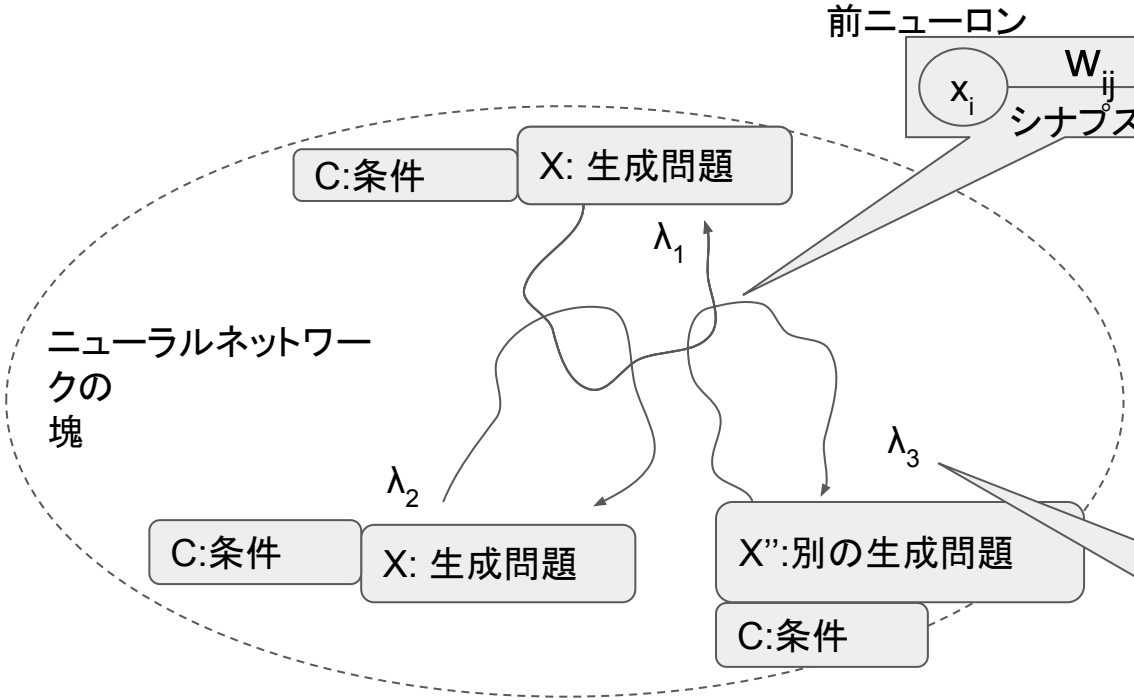
正則均衡勾配法 [Laborieux 2022]

均衡勾配法において、λを複素数に拡張する

- λが複素数であり、振動している場合、その最初の周波数成分がパラメータについての勾配に対応する

→ 推論しながら、同時に第一周波数成分をフィルタリング（フーリエ変換）することで、学習情報も得られ、推論と学習を同時に実現できる

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}W_{i,j}} = \frac{1}{T|\beta|}\int_0^T \rho(s^*_{\beta(t),i})\rho(s^*_{\beta(t),j})\exp(\frac{-2i\pi t}{T})\mathrm{d}t$$



様々な位置での教師あり学習（デノイジング）が同時多発的に行われている
　これらの教師損失への固定は並列に行われ、常に新しい状態に収束している

各重みは通常の値（これまでの平均）と差をとって評価し更新されていく

各λは振動しており、第一周波数成分を取り出すと勾配に対応し学習と推論を同じフェーズで行える

残されている問題

- エネルギベースモデルのアーキテクチャで、現在のSOTAモデルと同様の性能はまだ達成していない
  - 計算の対称性がモデル設計の大きな制約
    （例：TransformerでKとVが別のを使えない）
  - NNで成功している手法（正規化層）がそのまま使えない
- エネルギーベースモデルの学習が大きなデータセットで成功していない
  - 勾配の近似精度（もしくはバイアス）が問題

まとめ

まとめ

- 条件付け生成モデルとして拡散モデル、フローベースモデルを紹介した
- 非同期、並列に学習可能なエネルギーベースモデルを紹介した
- これらを組み合わせた、脳内で同時多発的、非同期、並列で様々な問題を扱うことが可能な学習モデルの仮説を提唱した

文献

[Zhang+ 2023] “Adding Conditional Control to Text-to-Image Diffusion Models”, arXiv:2302.05543
[Janner+ 2022] “Planning with Diffusion for Flexible Behavior Synthesis”, ICML 2022
[Sohl-Dickstein+ 2015] “Deep Unsupervised learning using nonequilibrium thermodynamics”, ICML 2015
[Song+ 2019] “Generative Modeling by Estimating Gradients of the Data Distribution”, NeurIPS 2019
[Ho+ 2020] “Denosing Diffusion Probabilistic Mdoels”, NeurIPS 2020
[Ho+ 2022] “High Definition Video Generation with Diffusion Models”, arXiv:2210.02303

[Vincent+ 2010] “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion”, JMLR 2010
[Hyvarinen+ 2005] “Estimatio of Non-Normalized Statistical Models by Score Matching”, JMLR, 2005
[Vincent+ 2011] “A Connection Between Score Matching and Denoising AutoEncoder”, Neural Computation 2011
[Song+ 2021] “Score-Based Generative Modeling through Stochastic Differential Equations”, ICLR 2021
[Anderson 1982] “Reverse-Time Diffusion Equation Models”, Stochastic Procsses and their Applications, 12(3): 313-326, 1982

[Karras+ 2022] “Elucidating the Design Space of Diffusion-Based Generative Models”, NeurIPS 2022
[Feller 1949] “On the theory of stochastic processes, with particular reference to applications”, In Proc. of Bekerly Symposium on Mathematical Statistics and Probability
[Xu+ 2022] “GeoDiff: a Geometric Diffusion Mdodel for Molecular Conformation Generation”, ICLR 2022
[Ho+ 2022] “Classifier-Free Diffusion Guidance”, arXiv:2207.12598
[Lipman+ 2022] “Flow Matching for Generative Modelisng”, arXiv:2210.02747
[Tong+ 2023] “Improving and generalizing flow-based generative models with minibatch optimal transport”, arXiv: 2302.00482
[Chen+ 2023] “Riemannian Flow Matching on General Geometries”, arXiv:2302.03660

[Lillicrap+ 2014] “Random feedback weights support learning in deep neural networks”, Nature Communications 7(1):1-10 2016
[Gomez+ 2022] “Interlocking Backpropagation: Improving depthwise model-parallelism”, JMLR 2022
[Ramsauer+ 2020] “Hopfield Networks is All you need”, arXiv:2008.02217
[Millidge+ 2023] “Backpropagation at the Infinitesimal Inference Limit of Energy-Based Models: Unifying Predictive Coding, Equilibrium Propagation, and Contrastive Hebbign Learning”, ICLR 2023
[Song 2020] “Can the brain do backpropagation?”, NeurIPS 2020
[Xie 2003] “Equivalence of backpropagation and contrastive Hebbign learning in a layered network.” Neural Computation 2003

[Whittinton 2017] “An approximation of the error backpropagation algorithm in a predictive coding network with local Hebbian synaptic plasticity”, Neural Computation 2017
[Scellier 2017] “Equilibrium propagtion: Bridging the gap between energy-based models and backpropagation”, Frontiers in Computational Neuroscience, 2017
[Laborieux 2022] “Holomorphic equilibrium propagation computes exact gradients through finite size oscillations”, NeurIPS 2022