生成モデルとベイズ脳仮説

これまでの章では，外界からの入力に対して，いかにしてネットワークが意味のある出力を生成するか，という問題を主に扱ってきた．こうしたモデルは知覚をモデル化している．改めて基礎的な点を確認すると，知覚（perception）とは、外界からの刺激を感覚受容器によって受け取り，それに意味を与える過程を指す．

一方で，知覚は結果（感覚入力）から外界に存在する潜在的な原因を推定する逆推論の過程であると捉える枠組みがあり，これを生成的知覚 (generative perception) と呼ぶ．

例えば外界は3次元なのに対し，網膜像は2次元であり，脳は不良設計問題を解かねばならない．

逆光学 (inverse optics)

生成的知覚の観点では，知覚とは逆問題を解くこと，とも捉えることができる．

生成的知覚は，生成モデル (generative model) と呼ばれるモデルを必要とする．このため，まず生成モデルについて説明をする．

観測データ（感覚入力）を $\mathbf{x}$ とし，その確率分布を $p_{\text{data}}(\mathbf{x})$ とする．$p_{\text{data}}(\cdot)$ が既知であれば，データを生成（サンプリング）できるが，ほとんどの場合で $p_{\text{data}}(\cdot)$ は未知である．ここで，パラメータ θ を伴う確率モデル $p_\theta(\cdot)$ を導入する．$p_\theta(\cdot)$ が $p_{\text{data}}(\cdot)$ を近似できれば，観測データに近いデータを $p_\theta(\cdot)$ に基づいて生成することが可能である．この $p_\theta(\cdot)$ が生成モデルであり，生成モデルを訓練するとは $p_\theta(\cdot)$ が $p_{\text{data}}(\cdot)$ を近似するようにパラメータ θ を調整することである．

外界の変数がすべて感覚入力として得られる状態，すなわち全て観測可能 (fully visible) であればよいが，基本的には部分的にのみ観測可能 (partially visible) である．観測できない変数を潜在変数 (latent variable) $\mathbf{z}$ とする．潜在変数に基づいて観測データが生成される過程をモデル化すると，

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x} \mid \mathbf{z}) p_\theta(\mathbf{z}) \tag{1}$$

となる．

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})}{p(\mathbf{x})}$$

主成分分析や独立成分分析も生成モデルと捉えることが可能である．
主成分分析は...
独立成分分析は...

変分ベイズ推論は

https://arxiv.org/abs/1901.07945

生成的知覚が脳でも行われているという仮説，すなわち脳は確率的推論に基づいて感覚情報を処理し、外界の状態を推定しているという仮説をベイズ脳仮説（The Bayesian Brain Hypothesis）と呼ぶ．

は、脳が確率的推論に基づいて感覚情報を処理し、外界の状態を推定しているという理論である。この仮説において、脳は世界に関する内部モデルを構築しており、そこに入力される不完全かつ雑音を含む感覚情報をもとに、ベイズの定理を用いて外界の隠れた原因を推測する。ベイズの定理は、ある観測 x が与えられたときに、その観測を引き起こしたと考えられる原因 z の確率を次のように与える：

$$p(z \mid x) = \frac{p(x \mid z) \cdot p(z)}{p(x)}.$$

ここで、$p(z \mid x)$ は観測 x をもとにした原因 z の事後確率、$p(x \mid z)$ は原因 z に基づいて観測される x の尤度、$p(z)$ は原因に対する事前確率、そして $p(x)$ は観測全体の周辺尤度である。この定理に従って、脳は感覚情報に対して最も妥当な解釈を与える原因を推定することになる。

脳内の知覚処理は、単に入力された情報を逐次的に処理するのではなく、過去の経験や学習によって形成された事前分布 $p(z)$ に基づいて、現在の感覚入力 x を統合的に解釈する。たとえば、視覚において曖昧な像が網膜に映った場合でも、脳はこれまでに得た視覚的知識を用いて、その像が何であるかを推測する。この過程では、感覚入力の不確かさに応じて尤度 $p(x \mid z)$ を評価し、それを既存の事前分布と統合することで、最終的な事後分布 $p(z \mid x)$ を得る。

このようなベイズ的推論の過程は、近年の予測符号化（predictive coding）の理論とも密接に関連している。予測符号化モデルにおいては、脳は高次の神経回路から低次の回路へと予測信号を送り、それと実際の感覚入力との間に生じる予測誤差を下から上へと伝播させる。この誤差が学習や推論の駆動源となり、内部モデルが更新される。数式で表すと、予測誤差は次のように定義される：

$$\epsilon = x - \hat{x}(z),$$

ここで $\hat{x}(z)$ は原因 z に基づく感覚入力の予測値である。脳はこの予測誤差 ϵ を最小化する方向に内部表現 z を更新することで、より正確な知覚や認知を実現している。これは、事後確率 $p(z \mid x)$ を最大化する（すなわち MAP 推定を行う）操作に相当する。

このような理論は、知覚だけでなく注意、意思決定、運動制御、学習など、さまざまな脳機能に適用可能であり、実際、神経科学の実験においてもベイズ的推論と整合する結果が多数報告されている。たとえば、期待された刺激に対して視覚野の神経活動が抑制される現象は、予測が成功し誤差が小さくなったことを反映していると解釈される。また、注意の効果は、事前分布 $p(z)$ の重みづけの変化として理解される。

以上のように、ベイズ脳仮説は、脳の情報処理を確率論的推論としてとらえることで、感覚から行動に至る広範な認知機能を統一的に説明する枠組みを提供している。脳は不確実性を内包する世界の中で、限られた情報をもとに最も妥当な仮説を選び、常にそれを更新し続けるベイズ推論器として機能しているのである。

Knill, David C., and Alexandre Pouget. 2004. “The Bayesian Brain: The Role of Uncertainty in Neural Coding and Computation.” Trends in Neurosciences 27 (12): 712–19.

エネルギーベースモデル

エネルギーベースモデルではネットワークの状態をスカラー値に変換するエネルギー関数 (あるいはコスト関数) を定義し，推論時と学習時の双方においてエネルギーを最小化するようにネットワークの状態を更新する (LeCun, Chopra, Hadsell, Ranzato, & Huang, 2006)．エネルギーベースモデルとしてはIsingモデルや(Amari-)Hopfieldモデル，Boltzmannマシン等が該当する．モデルの神経活動を$\mathbf{x} \in \mathbb{R}^n$，パラメータ$\theta$, （ポテンシャル）エネルギー関数 $E_\theta : \mathbb{R}^n \to \mathbb{R}$とすると，$\mathbf{x}$の分布はGibbs-Boltzmann分布を用いて次のように表せる．

$$p_\theta(\mathbf{x}) = \frac{\exp\left(-\beta E_\theta\left(\mathbf{x}\right)\right)}{Z_\theta} \tag{2}$$

ただし，Z_θは規格化定数であり，$Z_\theta = \int -\beta E_\theta\left(\mathbf{x}\right) d\mathbf{x}$ である．定義した任意の $E_\theta(\mathbf{x})$ を神経活動$\mathbf{x}$やパラメータθで微分することで，推論と学習ダイナミクスを定義できる (Fig. 3)．逆に神経活動のダイナミクスを積分することでエネルギーを定義することもできる (Isomura & Friston, 2020)．

Fig. 3. (上) エネルギー，神経活動の確率分布，推論・学習ダイナミクスの関係．簡単のため$\beta = 1$とした．いずれかを定義すれば他が導出できる．確率分布は直接保持されず，神経活動のダイナミクスによるサンプリングで表現される．（下）神経活動のダイナミクスからエネルギーと学習ダイナミクスを導出する例．

エネルギーベースモデル（energy-based model; EBM）と呼ばれる確率モデルの枠組みを取り上げる。

エネルギーベースモデルでは，系の状態に

EBMsは、入力データに対して**スカラー値のエネルギー（あるいはコスト）を割り当てる関数**を定義し、そのエネルギーを最小化するようにシステムの状態を決定・学習するという特徴を持つ。これは、ニューラルネットワークなどの高次元な状態空間における確率的な推論や学習に広く応用されている枠組みである \citep{lecun2006tutorial}。

エネルギーベースモデルでは、入力ベクトル $\mathbf{x} \in \mathbb{R}^d$ に対して、パラメータ θ に依存する**エネルギー関数** (energy function) $E_\theta : \mathbb{R}^d \to \mathbb{R}$ を定義する。このエネルギー関数は、ある状態 $\mathbf{x}$ の「好まし

さ」や「自然さ」を定量的に評価するものであり、エネルギーが小さいほどその状態がより実現しやすいと解釈される。

このようなエネルギー関数を用いて、状態 $\mathbf{x}$ の**確率密度関数** $p_\theta(\mathbf{x})$ を以下のように定義する：

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z_\theta} \tag{3}$$

ここで $Z_\theta := \int \exp(-E_\theta(\mathbf{x}))\, d\mathbf{x}$ は**分配関数**（partition function）と呼ばれ、確率分布を正規化するための定数である．Z_θ は状態空間全体にわたる積分であり、一般には計算が困難である点がEBMの学習と推論を難しくする主な要因である。

このように、エネルギーベースモデルは、確率分布を明示的にパラメトライズする代わりに、各状態に対するスカラーのスコア（＝エネルギー）を割り当て、そのスコアを通じて確率的な解釈を与えるという柔軟な表現力を持つ。そのため、EBMsは画像生成、異常検知、表現学習など、多様な応用分野で注目されている。

また、エネルギー関数 $E_\theta(\mathbf{x})$ の定義により、EBMs は生成モデルとして扱うこともできるが、識別的モデルとして利用することも可能である。たとえば、識別タスクにおいては、入力 $\mathbf{x}$ とラベル y のペア $(\mathbf{x}, y)$ に対して $E_\theta(\mathbf{x}, y)$ を定義し、正しいラベルに対するエネルギーを最小にするようなパラメータ θ を学習することができる。このように、EBM は生成と識別の両方の枠組みにまたがる柔軟なモデルである。

推論時と学習時の双方においてエネルギーを最小化するようにネットワークの状態を更新する

エネルギーベースモデルは、
神経系の状態遷移と安定性を記述する枠組みとして自然であり、
記憶や知覚といった脳の高次機能の数理的モデル化を可能にし、
確率的処理や最適化の観点からも神経活動の特徴をうまく表現できるため、
計算論的神経科学における理論的支柱の1つとして重要な役割を果たしています。

代謝コスト (metabolic cost) との関連．
脳は20Wしか消費しない？

ATP消費
https://www.cell.com/trends/cognitive-sciences/fulltext/S1364-6613(24)00319-X

エネルギーベースモデルでのエネルギーは代謝コストと一対一関係するものではない．
脳の代謝コストは、

ニューロンの発火活動

シナプス伝達

イオンポンプによる電位回復（たとえばNa/K ポンプ） などによって消費される**実際のエネルギー（ATPなど）**を指します。

定量的には、脳は体重の約2%しか占めないにも関わらず、体全体のエネルギー消費の20%前後を使う
その大部分は神経活動、特に**シナプス活動（興奮性シナプス）**に起因する
ことが知られています。これは物理的、化学的なエネルギー消費です。

ボルツマンマシン

エネルギーベースモデルの一種としてボルツマンマシン (Boltzmann machine) を取り上げる．

生成モデル (generative model)

Hopfieldモデルの各ユニットが取りうる活動を確率的にしたモデルがBoltzmannマシンである．

Boltzmannマシンは、確率的生成モデルの一例として、その状態の確率分布をエネルギー関数に基づいて定義するモデルである。ここで、システムの状態は $\mathbf{s} = (s_1, s_2, \ldots, s_N)$ という2値のユニットの組で表され、各 s_i は0または1の値を取る。Boltzmannマシンでは、各状態のエネルギーは以下の式によって与えられる：

$$E(\mathbf{s}) = -\sum_i b_i s_i - \sum_{i<j} W_{ij} s_i s_j$$

ここで、b_i は各ユニットに対応するバイアス項、W_{ij} はユニット i と j の間の対称的な結合重みを表す。状態 $\mathbf{s}$ が出現する確率は、エネルギー関数に基づいてボルツマン分布として定義され、以下のように記述される：

$$P(\mathbf{s}) = \frac{1}{Z} \exp\left(-E(\mathbf{s})\right)$$

ここで、正規化定数 Z（分配関数）は全状態にわたる和で定義される：

$$Z = \sum_{\mathbf{s}} \exp\left(-E(\mathbf{s})\right)$$

このモデルは、全ユニット間に結合が存在するため、内部の依存関係が複雑になり、特に学習の際にパラメータ更新のための勾配計算が指数的な計算量を要するという難点がある。

Boltzmannマシンにおける学習および推論の主要な困難さは、その計算に内在する分配関数 Z の評価に起因する。Boltzmannマシンでは、エネルギー関数

$$E(\mathbf{s}) = -\sum_{i} b_i s_i - \sum_{i<j} W_{ij} s_i s_j$$

に従い、状態 $\mathbf{s}$ の確率分布は

$$P(\mathbf{s}) = \frac{1}{Z} \exp\left(-E(\mathbf{s})\right)$$

と定義されるが、ここで正規化定数 Z は

$$Z = \sum_{\mathbf{s}} \exp\left(-E(\mathbf{s})\right)$$

と全可能状態 $\mathbf{s}$ にわたる和として計算されなければならない。各ユニットが2値の確率変数である場合、全状態数は 2^N となるため、ネットワークの規模が大きくなるとこの和は指数関数的に増大し、厳密な計算が事実上不可能となる。

さらに、学習に必要なパラメータ更新のための勾配計算でも、この正規化定数 Z に依存する項が現れる。具体的には、尤度関数の勾配として、例えば重み W_{ij} に関しては

$$\frac{\partial \log P(\mathbf{s})}{\partial W_{ij}} = \langle s_i s_j \rangle_{\text{data}} - \langle s_i s_j \rangle_{\text{model}}$$

と表されるが、ここで $\langle s_i s_j \rangle_{\text{model}}$ はモデル分布における期待値であり、これは

$$\langle s_i s_j \rangle_{\text{model}} = \sum_{\mathbf{s}} s_i s_j \, P(\mathbf{s})$$

として計算される必要がある。しかし、前述のように $P(\mathbf{s})$ の計算には Z の求積が不可欠であり、これもまた指数的な計算量を要するため、直接計算することは困難である。

このような計算の困難性は、統計物理における分配関数の計算問題と同様に、組み合わせ爆発（combinatorial explosion）の問題として知られ、計算複雑性理論では #P困難（#P-complete）であると指摘される。これに対処するため、実際の学習ではサンプルに基づく近似手法（モンテカルロ法、ギブスサンプリングなど）や、特定の近似アルゴリズム（コントラスト・ダイバージェンスなど）が利用される。しかしこれら近似手法にも収束の問題や精度の限界が存在するため、一般的なBoltzmannマシンは大規模な問題に対して直接適用するのが難しく、その計算効率の改善は依然として重要な研究課題である。

この問題点を解消するために考案されたのが、制限Boltzmannマシン（Restricted Boltzmann Machine: RBM）である。RBMでは、ネットワークを二層構造に限定し、可視層 $\mathbf{v}$ と隠れ層 $\mathbf{h}$ のみ

を用いる。ここで、可視ユニット v_i は入力データを表し、隠れユニット h_j はデータの特徴（潜在変数）を表す。RBMのエネルギー関数は次の形で定義される：

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i W_{ij} h_j$$

このとき、a_i は可視ユニットのバイアス、b_j は隠れユニットのバイアス、そして W_{ij} は可視ユニットと隠れユニット間の結合重みである。RBMでは、同一層内のユニット間の結合（例えば、可視層同士、隠れ層同士）は存在しないため、モデル内の条件付き独立性が成立する。具体的には、隠れ層の各ユニットは可視層が与えられた条件下で独立に分布し、その条件付き確率は次の式で表される：

$$P(h_j = 1 \mid \mathbf{v}) = \sigma\left(b_j + \sum_i v_i W_{ij}\right)$$

また、可視層の各ユニットに関しても同様に、

$$P(v_i = 1 \mid \mathbf{h}) = \sigma\left(a_i + \sum_j h_j W_{ij}\right)$$

と記述される。ここで、$\sigma(x) = \frac{1}{1+\exp(-x)}$ はシグモイド関数である。これらの性質により、RBMは効率的なギブスサンプリングが可能となり、コントラスト・ダイバージェンス（Contrastive Divergence, CD）と呼ばれる近似的な学習アルゴリズムが用いられて実用的な学習が可能となる。

このようにして、Boltzmannマシンは複雑な結合を持つモデルとして理論的な基盤を提供する一方、RBMはその結合を制限することにより計算の効率化を実現している。これらのモデルは、特にディープラーニングにおける事前学習や特徴抽出の文脈で重要な役割を果たし、画像認識や信号処理など幅広い応用がなされている。

制限ボルツマンマシン

(Restricted Boltzmann machine)
(cf.) http://deeplearning.net/tutorial/rbm.html

離散の観測変数(visible variable) $\mathbf{v}$, 潜在変数(hidden variable) $\mathbf{h}$とする．各ユニットの値は$\{0, 1\}$の2値 (binary)である．

エネルギー関数を

$$E_\theta(\mathbf{v}, \mathbf{h}) = -\mathbf{b}^\top \mathbf{v} - \mathbf{c}^\top \mathbf{h} + \mathbf{v}^\top \mathbf{W} \mathbf{h} \tag{4}$$

とする．ただし，$\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$

シグモイド関数を

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{5}$$

とする．

訓練データで学習

$$p_\theta(\mathbf{h}|\mathbf{v}) = \prod_i p_\theta(h_i = 1|\mathbf{v}) = \prod_i \sigma(c_i + W_i \mathbf{v}) \tag{6}$$

$$p_\theta(\mathbf{v}|\mathbf{h}) = \prod_j p_\theta(v_j = 1|\mathbf{h}) = \prod_j \sigma(b_j + W_j^\top \mathbf{h}) \tag{7}$$

階層的生成モデル

生成モデルの表現力を高めるため，生成モデルを階層化することを考えよう．

本章では階層的生成モデルを導入し，それからスパース符号化，予測符号化について説明する．

MAP推定で行う．

スパース符号化モデル

スパース符号化と生成モデル

スパース符号化モデル (Sparse coding model) \citep{`Olshausen1996-xe` \citep{`Olshausen1997-qu` はV1のニューロンの応答特性を説明する**線形生成モデル** (linear generative model)である．まず，画像パッチ $\mathbf{x}$ が基底関数(basis function) $\mathbf{\Phi} = [\phi_j]$ のノイズを含む線形和で表されるとする (係数は $\mathbf{r} = [r_j]$ とする)．

$$\mathbf{x} = \sum_j r_j \phi_j + \boldsymbol{\epsilon} = \mathbf{\Phi}\mathbf{r} + \boldsymbol{\epsilon} \tag{8}$$

ただし，$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ である．このモデルを神経ネットワークのモデルと考えると， $\mathbf{\Phi}$ は重み行列，係数 $\mathbf{r}$ は入力よりも高次の神経細胞の活動度を表していると解釈できる．ただし，r_j は負の値も取るので単純に発火率と捉えられないのはこのモデルの欠点である．

Sparse codingでは神経活動 $\mathbf{r}$ が潜在変数の推定量を表現しているという仮定の下，少数の基底で画像 (や目的変数)を表すことを目的とする．要は上式において，ほとんどが0で，一部だけ0以外の値を

取るという疎 (=sparse)な係数$\mathbf{r}$を求めたい．

確率的モデルの記述

入力される画像パッチ $\mathbf{x}_i \ (i = 1, \ldots, N)$ の真の分布を $p_{data}(\mathbf{x})$ とする．また，$\mathbf{x}$ の生成モデルを $p(\mathbf{x}|\mathbf{\Phi})$ とする．さらに潜在変数 $\mathbf{r}$ の事前分布 (prior)を $p(\mathbf{r})$, 画像パッチ $\mathbf{x}$ の尤度 (likelihood)を $p(\mathbf{x}|\mathbf{r}, \mathbf{\Phi})$ とする．このとき，

$$p(\mathbf{x}|\mathbf{\Phi}) = \int p(\mathbf{x}|\mathbf{r}, \mathbf{\Phi}) p(\mathbf{r}) d\mathbf{r} \tag{9}$$

が成り立つ．$p(\mathbf{x}|\mathbf{r}, \mathbf{\Phi})$は，(1)式においてノイズ項を$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$としたことから，

$$p(\mathbf{x}|\ \mathbf{r}, \mathbf{\Phi}) = \mathcal{N}\left(\mathbf{x}|\ \mathbf{\Phi}\mathbf{r}, \sigma^2\mathbf{I}\right) = \frac{1}{Z_\sigma} \exp\left(-\frac{\|\mathbf{x} - \mathbf{\Phi}\mathbf{r}\|^2}{2\sigma^2}\right) \tag{10}$$

と表せる．ただし，Z_σは規格化定数である．

事前分布の設定

事前分布$p(\mathbf{r})$としては，0においてピークがあり，裾の重い(heavy tail)を持つsparse distributionあるいは **super-Gaussian distribution** (Laplace分布やCauchy分布などGaussian分布よりもkurtoticな分布) を用いるのが良い．このような分布では，$\mathbf{r}$の各要素r_iはほとんど0に等しく，ある入力に対しては大きな値を取る．$p(\mathbf{r})$は一般化して次のように表記する．

$$p(\mathbf{r}) = \prod_j p(r_j) \tag{11}$$

$$p(r_j) = \frac{1}{Z_\beta} \exp\left[-\beta S(r_j)\right] \tag{12}$$

ただし，βは逆温度(inverse temperature), Z_βは規格化定数（分配関数）である．これらの用語は統計力学における正準分布 (Boltzmann分布)から来ている．$S(x)$と分布の関係をまとめた表が以下となる．

$S(r)$	$\frac{dS(r)}{dr}$	$p(r)$	分布名	尖度
r^2	$2r$	$\frac{1}{\alpha\sqrt{2\pi}} \exp\left(-\frac{r^2}{2\alpha^2}\right)$	Gaussian 分布	0
$\lvert r\rvert$	$\mathrm{sign}(r)$	$\frac{1}{2\alpha} \exp\left(-\frac{\lvert r\rvert}{\alpha}\right)$	Laplace 分布	3.0
$\ln(\alpha^2 + r^2)$	$\frac{2r}{\alpha^2 + r^2}$	$\frac{\alpha}{\pi}\frac{1}{\alpha^2 + r^2} = \frac{\alpha}{\pi}\exp[-\ln(\alpha^2 + r^2)]$	Cauchy 分布	$-$

分布$p(r)$や$S(r)$を描画すると次のようになる．

目的関数の設定と最適化

最適な生成モデルを得るために，入力される画像パッチの真の分布 $p_{data}(\mathbf{x})$と$\mathbf{x}$の生成モデル $p(\mathbf{x}|\mathbf{\Phi})$を近づける．このために，2つの分布のKullback-Leibler ダイバージェンス $D_{\mathrm{KL}}\left(p_{data}(\mathbf{x})\|\, p(\mathbf{x}|\mathbf{\Phi})\right)$を最小化したい．しかし，真の分布は得られないので，経験分布

$$\hat{p}_{data}(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}-\mathbf{x}_i) \tag{13}$$

を近似として用いる ($\delta(\cdot)$ はDiracのデルタ関数である)．ゆえに$D_{\mathrm{KL}}\left(\hat{p}_{data}(\mathbf{x})\|\, p(\mathbf{x}|\mathbf{\Phi})\right)$を最小化する．

$$\begin{aligned} D_{\mathrm{KL}}\left(\hat{p}_{data}(\mathbf{x})\|\, p(\mathbf{x}|\mathbf{\Phi})\right) &= \int \hat{p}_{data}(\mathbf{x})\log\frac{\hat{p}_{data}(\mathbf{x})}{p(\mathbf{x}|\mathbf{\Phi})}d\mathbf{x} && (14)\\ &= \mathbb{E}_{\hat{p}_{data}}\left[\ln\frac{\hat{p}_{data}(\mathbf{x})}{p(\mathbf{x}|\mathbf{\Phi})}\right] && (15)\\ &= \mathbb{E}_{\hat{p}_{data}}\left[\ln\hat{p}_{data}(\mathbf{x})\right] - \mathbb{E}_{\hat{p}_{data}}\left[\ln p(\mathbf{x}|\mathbf{\Phi})\right] && (16)\end{aligned}$$

が成り立つ．(7)式の1番目の項は一定なので，$D_{\mathrm{KL}}\left(\hat{p}_{data}(\mathbf{x})\|\, p(\mathbf{x}|\mathbf{\Phi})\right)$ を最小化するには $\mathbb{E}_{\hat{p}_{data}}\left[\ln p(\mathbf{x}|\mathbf{\Phi})\right]$を最大化すればよい．ここで，

$$\mathbb{E}_{\hat{p}_{data}}\left[\ln p(\mathbf{x}|\mathbf{\Phi})\right] = \sum_{i=1}^{N}\hat{p}_{data}(\mathbf{x}_i)\ln p(\mathbf{x}_i|\mathbf{\Phi}) = \frac{1}{N}\sum_{i=1}^{N}\ln p(\mathbf{x}_i|\mathbf{\Phi}) \tag{17}$$

が成り立つ．また，(2)式より

$$\ln p(\mathbf{x}|\mathbf{\Phi}) = \ln\int p(\mathbf{x}|\mathbf{r},\mathbf{\Phi})p(\mathbf{r})d\mathbf{r} \tag{18}$$

が成り立つので，近似として $\displaystyle\int p(\mathbf{x}|\mathbf{r},\mathbf{\Phi})p(\mathbf{r})d\mathbf{r}$ を $p(\mathbf{x}|\mathbf{r},\mathbf{\Phi})p(\mathbf{r})\ (=p(\mathbf{x},\mathbf{r}|\mathbf{\Phi}))$ で評価する．
これらの近似の下，最適な$\mathbf{\Phi}=\mathbf{\Phi}^*$は次のようにして求められる．

$$\begin{aligned}
\mathbf{\Phi}^* &= \arg\min_{\mathbf{\Phi}} \min_{\mathbf{r}} D_{\mathrm{KL}}\left(\hat{p}_{data}(\mathbf{x}) \| p(\mathbf{x}|\mathbf{\Phi})\right) && (19)\\
&= \arg\max_{\mathbf{\Phi}} \max_{\mathbf{r}} \mathbb{E}_{\hat{p}_{data}}\left[\ln p(\mathbf{x}|\mathbf{\Phi})\right] && (20)\\
&= \arg\max_{\mathbf{\Phi}} \sum_{i=1}^{N} \max_{\mathbf{r}_i} \ln p(\mathbf{x}_i|\mathbf{\Phi}) && (21)\\
&\approx \arg\max_{\mathbf{\Phi}} \sum_{i=1}^{N} \max_{\mathbf{r}_i} \ln p(\mathbf{x}_i|\mathbf{r}_i, \mathbf{\Phi}) p(\mathbf{r}_i) && (22)\\
&= \arg\min_{\mathbf{\Phi}} \sum_{i=1}^{N} \min_{\mathbf{r}_i} \ E(\mathbf{x}_i, \mathbf{r}_i|\mathbf{\Phi}) && (23)
\end{aligned}$$

ただし，$\mathbf{x}_i$に対する神経活動を $\mathbf{r}_i$とした．また，$E(\mathbf{x}, \mathbf{r}|\mathbf{\Phi})$はコスト関数であり，次式のように表される．

$$\begin{aligned}
E(\mathbf{x}, \mathbf{r}|\mathbf{\Phi}) &:= -\ln p(\mathbf{x}|\mathbf{r}, \mathbf{\Phi}) p(\mathbf{r}) && (24)\\
&= \underbrace{\|\mathbf{x} - \mathbf{\Phi}\mathbf{r}\|^2}_{\text{preserve information}} + \lambda \underbrace{\sum_j S\left(r_j\right)}_{\text{sparseness of } r_j} && (25)
\end{aligned}$$

ただし，$\lambda = 2\sigma^2\beta$は正則化係数(この式から逆温度βが正則化の度合いを調整するパラメータであることがわかる.)であり，1行目から2行目へは式(3), (4), (5)を用いた．ここで，第1項が復元損失，第2項が罰則項 (正則化項)となっている．

式(9)で表される最適化手順を最適な$\mathbf{r}$と$\mathbf{\Phi}$を求める過程に分割しよう．まず， $\mathbf{\Phi}$を固定した下で$E(\mathbf{x}_n, \mathbf{r}_i|\mathbf{\Phi})$を最小化する$\mathbf{r}_i = \hat{\mathbf{r}}_i$を求める．

$$\hat{\mathbf{r}}_i = \arg\min_{\mathbf{r}_i} E(\mathbf{x}_i, \mathbf{r}_i|\mathbf{\Phi}) \ \left(= \arg\max_{\mathbf{r}_i} p(\mathbf{r}_i|\mathbf{x}_i)\right) \tag{26}$$

これは $\mathbf{r}$ について **MAP推定** (maximum a posteriori estimation)を行うことに等しい．次に$\hat{\mathbf{r}}$を用いて

$$\mathbf{\Phi}^* = \arg\min_{\mathbf{\Phi}} \sum_{i=1}^{N} E(\mathbf{x}_i, \hat{\mathbf{r}}_i|\mathbf{\Phi}) \ \left(= \arg\max_{\mathbf{\Phi}} \prod_{i=1}^{N} p(\mathbf{x}_i|\hat{\mathbf{r}}_i, \mathbf{\Phi})\right) \tag{27}$$

とすることにより，$\mathbf{\Phi}$を最適化する．こちらは $\mathbf{\Phi}$ について **最尤推定** (maximum likelihood estimation)を行うことに等しい．

局所競合則

局所競合則 (Locally competitive algorithm; LCA).

$\mathbf{r}$の勾配法による更新則は，Eの微分により次のように得られる．

$$\frac{d\mathbf{r}}{dt} = -\frac{\eta_{\mathbf{r}}}{2}\frac{\partial E}{\partial \mathbf{r}} = \eta_{\mathbf{r}} \cdot \left[\mathbf{\Phi}^{\top}(\mathbf{x} - \mathbf{\Phi}\mathbf{r}) - \frac{\lambda}{2}S'(\mathbf{r})\right] \tag{28}$$

ただし，$\eta_{\mathbf{r}}$は学習率である．この式により$\mathbf{r}$が収束するまで最適化するが，単なる勾配法ではなく，\citep{ `Olshausen1996-xe` では**共役勾配法** (conjugate gradient method)を用いている．しかし，共役勾配法は実装が煩雑で非効率であるため，より効率的かつ生理学的な妥当性の高い学習法として，**LCA** (locally competitive algorithm)が提案されている \citep{ `Rozell2008-wp` ．LCAは**側抑制** (local competition, lateral inhibition)と**閾値関数** (thresholding function)を用いる更新則である．LCAによる更新を行うRNNは通常のRNNとは異なり，コスト関数(またはエネルギー関数)を最小化する動的システムである．このような機構はHopfield networkで用いられているために，Olshausenは**Hopfield trick**と呼んでいる．

軟判定閾値関数を用いる場合 (ISTA)

$S(x) = |x|$とした場合の閾値関数を用いる手法として**ISTA**(Iterative Shrinkage Thresholding Algorithm)がある．ISTAはL1-norm正則化項に対する近接勾配法で，要はLasso回帰に用いる勾配法である．

解くべき問題は次式で表される．

$$\mathbf{r} = \arg\min_{\mathbf{r}} \left\{\|\mathbf{x} - \mathbf{\Phi}\mathbf{r}\|_2^2 + \lambda\|\mathbf{r}\|_1\right\} \tag{29}$$

詳細は後述するが，次のように更新することで解が得られる．

- $\mathbf{r}(0)$を要素が全て0のベクトルで初期化：$\mathbf{r}(0) = \mathbf{0}$
- $\mathbf{r}_*(t+1) = \mathbf{r}(t) + \eta_{\mathbf{r}} \cdot \mathbf{\Phi}^{\top}(\mathbf{x} - \mathbf{\Phi}\mathbf{r}(t))$
- $\mathbf{r}(t+1) = \Theta_{\lambda}(\mathbf{r}_*(t+1))$
- $\mathbf{r}$が収束するまで2と3を繰り返す

ここで$\Theta_{\lambda}(\cdot)$は**軟判定閾値関数** (Soft thresholding function)と呼ばれ，次式で表される．

$$\Theta_{\lambda}(y) = \begin{cases} y - \lambda & (y > \lambda) \\ 0 & (-\lambda \leq y \leq \lambda) \\ y + \lambda & (y < -\lambda) \end{cases} \tag{30}$$

$\Theta_{\lambda}(\cdot)$を関数として定義すると次のようになる．また，ReLU (ランプ関数)は `max(x, 0)` で実装できる．この点から考えればReLUを軟判定非負閾値関数 (soft nonnegative thresholding function)と捉えることもできる \citep{ `Papyan2018-yr` ．

なお，軟判定閾値関数は次の目的関数Cを最小化するxを求めることで導出できる．

$$C = \frac{1}{2}(y-x)^2 + \lambda|x| \tag{31}$$

ただし，x, y, λはスカラー値とする．$|x|$が微分できないが，これは場合分けを考えることで解決する．$x \geq 0$を考えると，(6)式は

$$C = \frac{1}{2}(y-x)^2 + \lambda x = \{x-(y-\lambda)\}^2 + \lambda(y-\lambda) \tag{32}$$

となる．(7)式の最小値を与えるxは場合分けをして考えると，$y-\lambda \geq 0$のとき二次関数の頂点を考えて$x = y-\lambda$となる． 一方で$y-\lambda < 0$のときは$x \geq 0$において単調増加な関数となるので，最小となるのは$x=0$のときである．同様の議論を$x \leq 0$に対しても行うことで (5)式が得られる．

なお，閾値関数としては軟判定閾値関数だけではなく，硬判定閾値関数や$y = x - \tanh(x)$ (Tanh-shrink)など様々な関数を用いることができる．

重み行列の更新則

$\mathbf{r}$が収束したら勾配法により$\mathbf{\Phi}$を更新する．

$$\Delta\phi_i(\boldsymbol{x}) = -\eta\frac{\partial E}{\partial \mathbf{\Phi}} = \eta \cdot \left[(\mathbf{x} - \mathbf{\Phi r})\,\mathbf{r}^\top\right] \tag{33}$$

スパース符号化モデルの実装

ネットワークは入力層を含め2層の単純な構造である．今回は，入力はランダムに切り出した16×16 (＝256)の画像パッチとし，これを入力層の256個のニューロンが受け取るとする．入力層のニューロンは次層の100個のニューロンに投射するとする．100個のニューロンが入力をSparseに符号化するようにその活動および重み行列を最適化する．

予測符号化モデル

u を w に変更．

Annotated Bibliographyはもう一度確認する．

Pece, AEC (1992) Redundancy reduction of a Gabor representation: A possible computational role for feedback from primary visual cortex to lateral geniculate nucleus. In I Aleksander, & J Taylor, eds., Artificial Neural Networks, 2, 865–868. Amsterdam: Elsevier

Kawato, M, Hayakama, H, & Inui, T (1993) A forward-inverse optics model of reciprocal connections between visual cortical areas. Network: Computation in Neural Systems 4:415–422.

https://arxiv.org/abs/2011.07464
https://arxiv.org/abs/2112.10048

https://arxiv.org/abs/2410.19315
Predictive coding as variational inference

Srinivasan, M. V., Laughlin, S., & Dubs, A. (1982). Predictive coding: a fresh view of inhibition in the retina. Proceedings of the Royal Society of London. Series B. Biological Sciences, 216(1205), 427–459.

Dong, D. W., & Atick, J. J. (1995). Temporal decorrelation: a theory of lagged and nonlagged responses in the lateral geniculate nucleus. Network: Computation in Neural Systems, 6(2), 159–178.

A forward-inverse optics model of reciprocal connections between visual cortical areas
https://www.tandfonline.com/doi/abs/10.1088/0954-898X_4_4_001

https://pmc.ncbi.nlm.nih.gov/articles/PMC1569488/#bib45
https://pubmed.ncbi.nlm.nih.gov/15937014/

https://arxiv.org/abs/2212.00720

Kalman filter

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 82(1), 35–45.

観測世界の階層的予測

階層的予測符号化(hierarchical predictive coding; HPC) は\citep{ Rao1999-zv により導入された．構築するネットワークは入力層を含め，3層のネットワークとする．LGNへの入力として画像 $\mathbf{x} \in \mathbb{R}^{n_0}$ を考える．画像 $\mathbf{x}$ の観測世界における隠れ変数，すなわち**潜在変数** (latent variable) を $\mathbf{r} \in \mathbb{R}^{n_1}$ とし，ニューロン群によって発火率で表現されているとする (真の変数と $\mathbf{r}$ は異なるので文字を分けるべきだが簡単のためにこう表す)．このとき，

$$\mathbf{x} = f(\mathbf{U}\mathbf{r}) + \boldsymbol{\epsilon} \tag{34}$$

が成立しているとする．ただし，$f(\cdot)$は活性化関数 (activation function)，$\mathbf{U} \in \mathbb{R}^{n_0 \times n_1}$は重み行列である．$\boldsymbol{\epsilon} \in \mathbb{R}^{n_0}$ は $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ からサンプリングされるとする．潜在変数 $\mathbf{r}$はさらに高次 (higher-level)の潜在変数 $\mathbf{r}^h$により，次式で表現される．

$$\mathbf{r} = \mathbf{r}^{td} + \boldsymbol{\epsilon}^{td} = f(\mathbf{U}^h\mathbf{r}^h) + \boldsymbol{\epsilon}^{td} \tag{35}$$

ただし，Top-downの予測信号を $\mathbf{r}^{td} := f(\mathbf{U}^h \mathbf{r}^h)$ とした．また，$\mathbf{r}^{td} \in \mathbb{R}^{n_1}$, $\mathbf{r}^h \in \mathbb{R}^{n_2}$, $\mathbf{U}^h \in \mathbb{R}^{n_1 \times n_2}$ である．$\boldsymbol{\epsilon}^{td} \in \mathbb{R}^{n_1}$は$\mathcal{N}(\mathbf{0}, \sigma_{td}^2 \mathbf{I})$ からサンプリングされるとする．

話は飛ぶが，Predictive codingのネットワークの特徴は

- 階層的な構造
- 高次による低次の予測 (Feedback or Top-down信号)
- 低次から高次への誤差信号の伝搬 (Feedforward or Bottom-up 信号)

である．ここまでは高次表現による低次表現の予測，というFeedback信号について説明してきたが，この部分はSparse codingでも同じである．それではPredictive codingのもう一つの要となる，低次から高次への予測誤差の伝搬というFeedforward信号はどのように導かれるのだろうか．結論から言えば，これは復元誤差 (reconstruction error)の最小化を行う再帰的ネットワーク (recurrent network)を考慮することで自然に導かれる．

損失関数と学習則

事前分布の設定

$\mathbf{r}$の事前分布$p(\mathbf{r})$はCauchy分布を用いる．$p(\mathbf{r})$の負の対数事前分布を$g(\mathbf{r}) := -\log p(\mathbf{r})$としておく．

$$
p(\mathbf{r}) = \prod_i p(r_i) = \prod_i \exp\left[-\alpha \ln(1 + r_i^2)\right] \tag{36}
$$

$$
g(\mathbf{r}) = -\ln p(\mathbf{r}) = \alpha \sum_i \ln(1 + r_i^2) \tag{37}
$$

$$
g'(\mathbf{r}) = \frac{\partial g(\mathbf{r})}{\partial \mathbf{r}} = \left[\frac{2\alpha r_i}{1 + r_i^2}\right]_i \tag{38}
$$

次に重み行列$\mathbf{U}$の事前分布 $p(\mathbf{U})$はGaussian分布とする．$p(\mathbf{U})$の負の対数事前分布を$h(\mathbf{U}) := -\ln p(\mathbf{U})$とすると，次のように表される．

$$
p(\mathbf{U}) = \exp(-\lambda \|\mathbf{U}\|_F^2) \tag{39}
$$

$$
h(\mathbf{U}) = -\ln p(\mathbf{U}) = \lambda \|\mathbf{U}\|_F^2 \tag{40}
$$

$$
h'(\mathbf{U}) = \frac{\partial h(\mathbf{U})}{\partial \mathbf{U}} = 2\lambda \mathbf{U} \tag{41}
$$

ただし，$\|\cdot\|_F^2$はフロベニウスノルムを意味する．

損失関数の設定

Sparse codingと同様に考えることにより，損失関数 Eを次のように定義する．

$$E = \underbrace{\frac{1}{\sigma^2}\|\mathbf{x} - f(\mathbf{U}\mathbf{r})\|^2 + \frac{1}{\sigma_{td}^2}\left\|\mathbf{r} - f(\mathbf{U}^h\mathbf{r}^h)\right\|^2}_{\text{reconstruction error}} + \underbrace{g(\mathbf{r}) + g(\mathbf{r}^h) + h(\mathbf{U}) + h(\mathbf{U}^h)}_{\text{sparsity penalty}} \quad (42)$$

潜在変数 $\mathbf{r}, \mathbf{r}^h$ と 重み行列 $\mathbf{U}, \mathbf{U}^h$ のそれぞれに事前分布を仮定しているため，これらについてのMAP推定を行うことに相当する．

再帰ネットワークの更新則

簡単のために$\mathbf{z} := \mathbf{U}\mathbf{r}, \mathbf{z}^h := \mathbf{U}^h\mathbf{r}^h$とする．

$$\frac{d\mathbf{r}}{dt} = -\frac{k_1}{2}\frac{\partial E}{\partial \mathbf{r}} = k_1 \cdot \left(\frac{1}{\sigma^2}\mathbf{U}^\top\left[\frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} \odot \underbrace{(\mathbf{x} - f(\mathbf{z}))}_{\text{bottom-up error}}\right] - \frac{1}{\sigma_{td}^2}\underbrace{(\mathbf{r} - f(\mathbf{z}^h))}_{\text{top-down error}} - \frac{1}{2}g'(\mathbf{r})\right) \quad (43)$$

$$\frac{d\mathbf{r}^h}{dt} = -\frac{k_1}{2}\frac{\partial E}{\partial \mathbf{r}^h} = k_1 \cdot \left(\frac{1}{\sigma_{td}^2}(\mathbf{U}^h)^\top\left[\frac{\partial f(\mathbf{z}^h)}{\partial \mathbf{z}^h} \odot \underbrace{(\mathbf{r} - f(\mathbf{z}^h))}_{\text{bottom-up error}}\right] - \frac{1}{2}g'(\mathbf{r}^h)\right) \quad (44)$$

ただし，k_1は更新率 (updating rate)である．または，発火率の時定数を$\tau := 1/k_1$として，k_1は発火率の時定数τの逆数であると捉えることもできる．ここで1番目の式において，中間表現 $\mathbf{r}$ のダイナミクスはbottom-up errorとtop-down errorで記述されている．このようにbottom-up errorが $\mathbf{r}$ への入力となることは自然に導出される．なお，top-down errorに関しては高次からの予測 (prediction)の項$f(\mathbf{x}^h)$とleaky-integratorとしての項 $-\mathbf{r}$に分割することができる．また$\mathbf{U}^\top, (\mathbf{U}^h)^\top$は重み行列の転置となっており，bottom-upとtop-downの投射において対称な重み行列を用いることを意味している．$-g'(\mathbf{r})$は発火率を抑制してスパースにすることを目的とする項だが，無理やり解釈をすると自己再帰的な抑制と言える．

予測符号化による活動と結合の共調整

本節では予測符号化による

予測符号化による訓練

PCには"Standard" Generative PC と "Reverse" Discriminative PCが存在する．
Millidge, B., Seth, A., & Buckley, C. L. (2021). Predictive Coding: a Theoretical and Experimental Review. In arXiv [cs.AI]. arXiv. http://arxiv.org/abs/2107.12979

ここでのPCは"Reverse" Discriminative PC

状態をdecayすることで，generativeにもdiscriminativeにもすることが可能．
A Predictive-Coding Network That Is Both Discriminative and Generative

https://direct.mit.edu/neco/article/32/10/1836/95621/A-Predictive-Coding-Network-That-Is-Both

入出力を固定 (clamp) する．電位固定法のようなものか？predictive codingと文字を合わせる．(Song et al., 2023)

$x_0 = s_{in}, x_{L+1} = s_{target}$とする．状態$x_l(t=0) = \mathbf{0}(l = 2, \ldots, L)$に初期化する．予測誤差 ϵ_l を次式で計算する．

$$\epsilon_l(t) = \mathbf{z}_l(t) - \mathbf{w}_{l-1} f(\mathbf{z}_{l-1}(t)) \quad (l = 1, \ldots, L) \tag{45}$$

次に状態 $\mathbf{z}_l(t)$ $(t = 0, \ldots, \mathcal{T} - 1)$ を次式で更新する．

$$\mathbf{z}_l(t+1) = \mathbf{z}_l(t) + \gamma(-\epsilon_l + f'(\mathbf{z}_l(t))) \circ (\mathbf{w}_l^\top \epsilon_{l+1}(t)) \tag{46}$$

収束後，重みを次式で更新する．nを一つのsampleの番号として，

$$\mathbf{w}_l(n+1) = \mathbf{w}_l(n) + \eta \epsilon_l(\mathcal{T}) f(\mathbf{z}_l(\mathcal{T}))^\top \tag{47}$$

として重みを更新する．

順伝播 (forward propagation)

$f(\cdot)$を活性化関数とする．順伝播(feedforward propagation)は以下のようになる．$(\ell = 1, \ldots, L)$

$$\text{入力層}: \mathbf{z}_1 = \mathbf{x} \tag{48}$$
$$\text{隠れ層}: \mathbf{u}_\ell = W_\ell \mathbf{z}_\ell + \mathbf{b}_\ell \tag{49}$$
$$\mathbf{z}_{\ell+1} = f_\ell(\mathbf{u}_\ell) \tag{50}$$
$$\text{出力層}: \hat{\mathbf{y}} = \mathbf{z}_{L+1} \tag{51}$$

予測符号化による訓練

入出力を固定 (clamp) する．電位固定法のようなものか？predictive codingと文字を合わせる．(Rosebvbaum 2022, Song et al., 2023)

Rosenbaum, R. (2022). On the relationship between predictive coding and backpropagation. PloS One, 17(3), e0266102.

固定点解析によりbackpropと同等であることがわかる．
$\mathbf{z}_1 = \mathbf{x}_{\text{in}}, \mathbf{z}_{L+1} = \mathbf{x}_{\text{target}}$とする．状態$\mathbf{z}_\ell(t=0) = \mathbf{0}$ $(\ell = 2, \ldots, L)$に初期化する．予測誤差 $\boldsymbol{\epsilon}_\ell(t)$ を次式で計算する．

$$\boldsymbol{\epsilon}_\ell(t) = \mathbf{z}_{\ell+1}(t) - \mathbf{W}_\ell f(\mathbf{z}_\ell(t)) \quad (\ell = 1, \ldots, L-1) \tag{52}$$

$$\boldsymbol{\epsilon}_L = \frac{\partial \mathcal{L}(\mathbf{z}_{L+1}, \mathbf{x}_{\text{target}})}{\partial \mathbf{z}_{L+1}}$$

次に状態 $\mathbf{z}_\ell(t) \quad (\ell = 2, \ldots, L;\ t = 0, \ldots, \mathcal{T}-1)$ を次式で更新する．

$$\mathbf{z}_\ell(t+1) = \mathbf{z}_\ell(t) + \gamma(-\boldsymbol{\epsilon}_{\ell-1} + f'(\mathbf{z}_\ell(t))) \circ (\mathbf{w}_\ell^\top \boldsymbol{\epsilon}_\ell(t)) \tag{53}$$

収束後，重みを次式で更新する．nを一つのsampleの番号として，

$$\mathbf{w}_l(n+1) = \mathbf{w}_l(n) + \eta \epsilon_l(\mathcal{T}) f(\mathbf{z}_l(\mathcal{T}))^\top \tag{54}$$

として重みを更新する．

fixed prediction assumptionという (Millidge etal., 2022. Rosebvbaum 2022) 修正もある．

$$\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}_{L+1}} \tag{55}$$

$$\boldsymbol{\delta}_L := \frac{\partial \mathcal{L}}{\partial \mathbf{u}_L} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}_{L+1}} \frac{\partial \mathbf{z}_{L+1}}{\partial \mathbf{u}_L} \tag{56}$$

$$\boldsymbol{\delta}_\ell := \frac{\partial \mathcal{L}}{\partial \mathbf{u}_\ell} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}_{\ell+1}} \frac{\partial \mathbf{z}_{\ell+1}}{\partial \mathbf{u}_\ell} \tag{57}$$

$$= \left(\frac{\partial \mathcal{L}}{\partial \mathbf{u}_{\ell+1}} \frac{\partial \mathbf{u}_{\ell+1}}{\partial \mathbf{z}_{\ell+1}} \right) \frac{\partial \mathbf{z}_{\ell+1}}{\partial \mathbf{u}_\ell} \tag{58}$$

$$= {\mathbf{W}_{\ell+1}}^\top \boldsymbol{\delta}_{\ell+1} \odot f'_\ell(\mathbf{u}_\ell) \tag{59}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_\ell} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}_\ell} \frac{\partial \mathbf{z}_\ell}{\partial \mathbf{u}_\ell} \frac{\partial \mathbf{u}_\ell}{\partial \mathbf{W}_\ell} = \boldsymbol{\delta}_\ell \mathbf{z}_\ell^\top \tag{60}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}_\ell} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}_\ell} \frac{\partial \mathbf{z}_\ell}{\partial \mathbf{u}_\ell} \frac{\partial \mathbf{u}_\ell}{\partial \mathbf{b}_\ell} = \boldsymbol{\delta}_\ell \tag{61}$$

ベイズ線形回帰

ベイズ線形回帰 (Bayesian linear regression)
共役事前分布 (conjugate prior) を

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \tag{62}$$

と定義し，事後分布 (posterior) を

$$p(\mathbf{w}|\mathbf{Y}, \mathbf{X}) = \mathcal{N}(\mathbf{w}|\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) \tag{63}$$

とする．ただし，

$$\hat{\boldsymbol{\Sigma}}^{-1} = \boldsymbol{\Sigma}_0^{-1} + \beta \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \tag{64}$$

$$\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\Sigma}}(\beta \boldsymbol{\Phi}^\top \mathbf{y} + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0) \tag{65}$$

である．また，$\boldsymbol{\Phi} = \phi.(\mathbf{x})$であり，$\phi(x) = [1, x, x^2, x^3]$, $\boldsymbol{\mu}_0 = \mathbf{0}, \boldsymbol{\Sigma}_0 = \alpha^{-1}\mathbf{I}$とする．テストデータを$\mathbf{x}^*$とした際，予測分布は

$$p(y^*|\mathbf{x}^*, \mathbf{Y}, \mathbf{X}) = \mathcal{N}(y^*|\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \tag{66}$$

となる．ただし，

$$\boldsymbol{\mu}^* = \hat{\boldsymbol{\mu}}^\top \phi(\mathbf{x}^*) \tag{67}$$

$$\boldsymbol{\Sigma}^* = \frac{1}{\beta} + \phi(\mathbf{x}^*)^\top \hat{\boldsymbol{\Sigma}} \phi(\mathbf{x}^*) \tag{68}$$

である．

マルコフ連鎖モンテカルロ法

マルコフ連鎖モンテカルロ法 (MCMC)

前節では解析的に事後分布の計算をした．事後分布を近似的に推論する方法の1つに**マルコフ連鎖モンテカルロ法 (Markov chain Monte Carlo methods; MCMC)** がある．他の近似推論の手法としてはLaplace近似や変分推論（variational inference）などがある．MCMCは他の手法に比して，事後分布の推論だけでなく，確率分布を神経活動で表現する方法を提供するという利点がある．

データをXとし，パラメータをθとする．

$$p(\theta \mid X) = \frac{p(X \mid \theta) p(\theta)}{\int p(X \mid \theta) p(\theta) d\theta} \tag{69}$$

分母の積分計算$\int p(X \mid \theta) p(\theta) d\theta$が求まればよい．

エネルギーベースモデルとサンプリング

ポテンシャルエネルギー関数Eを下に凸の曲面，高次元の神経活動$\mathbf{x}$をその曲面を転がる球としよう．エネルギーの最小化に勾配降下を用いるエネルギーベースモデルでは球は斜面の勾配に沿って運動し，最小のエネルギー状態に到達する．Hopfieldモデルは単なる勾配降下であり，単純な勾配降下を用いるために極小解に陥りやすい．このために各ニューロンが確率的に0,1の値を取るBoltzmannマシンが考案された(Ackley, Hinton, & Sejnowski, 1985)．(制限)BoltzmannマシンではGibbsサンプリングを用い，各ユニットの活動を決める．制限Boltzmannマシンの問題点としては隠れ層間における結

合を認めないため感覚入力の無い自発発火を仮定できない点にある．よりモデル構築の柔軟性が高い発火率モデルあるいはspikingモデルにおけるRNNにおいて効率的にサンプリングを行うには，ノイズや振動を用いる (Fig. 4)．なお，点推定を行うには収束時に一定の発火率を保ち続ける必要があり，難しいと考えられる．

Fig. 4. 勾配法と勾配法にノイズ，振動を加えた場合の神経活動のダイナミクスの違い．（左上）2つの細胞の活動x_1，x_2に対するポテンシャルエネルギー．（右上段）ポテンシャルエネルギー局面上の神経活動の変化．左から勾配法，Langevinダイナミクス，Hamiltonian (+Langevin)ダイナミクス．（右下段）各ダイナミクスにおけるx_1，x_2の経時的変化．Hamiltonianダイナミクスでは振動（+ノイズ）を用いて効率的にサンプリングしている．

Boltzmanマシンでも使用した〜などとする．

ベイズ線形回帰

モンテカルロ法

マルコフ連鎖

Metropolis-Hastings法

ランジュバン・モンテカルロ法 (LMC)

拡散過程

$$\frac{d\theta}{dt} = \nabla \log p(\theta) + \sqrt{2} dW \tag{70}$$

Euler–Maruyama法により，

ハミルトニアン・モンテカルロ法 (HMC法)

LMCよりも一般的なMCMCの手法としてHamiltonianモンテカルロ法(Hamiltonian Monte Calro; HMC)あるいはハイブリッド・モンテカルロ法(Hybrid Monte Calro)がある．エネルギーポテンシャルの局面上をHamilton力学に従ってパラメータを運動させることにより高速にサンプリングする手法である．

一般化座標を$\mathbf{q}$, 一般化運動量を$\mathbf{p}$とする．ポテンシャルエネルギーを$U(\mathbf{q})$としたとき，古典力学（解析力学）において保存力のみが作用する場合の**ハミルトニアン (Hamiltonian)** $\mathcal{H}(\mathbf{q}, \mathbf{p})$は

$$\mathcal{H}(\mathbf{q}, \mathbf{p}) := U(\mathbf{q}) + \frac{1}{2}\|\mathbf{p}\|^2 \tag{71}$$

となる．このとき，次の2つの方程式が成り立つ．

$$\frac{d\mathbf{q}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} = \mathbf{p}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = -\frac{\partial U}{\partial \mathbf{q}} \tag{72}$$

これを**ハミルトンの運動方程式(hamilton's equations of motion)** あるいは**正準方程式 (canonical equations)** という．

リープフロッグ(leap frog)法により離散化する．

1. **共役事前分布を用いた解析的（閉形式）解**
   - ノイズがガウス，かつ回帰係数に対して共役なガウス事前分布を仮定すると，事後分布もガウスとなり，平均・分散を閉形式で得られる．
   - 具体的には，

     $$p(\boldsymbol{\beta} \mid X, y) = \mathcal{N}\big(\Sigma_n(X^T X)\beta_0 + \Sigma_n X^T y,\ \Sigma_n\big), \quad \Sigma_n = (X^T X + \Sigma_0^{-1})^{-1},$$

     のように書ける（PRML より） (Bayesian linear regression - Wikipedia)。
2. **ラプラス近似（Laplace’s method）**
   - 事後分布を最尤解（MAP）まわりの２次多項展開でガウス近似する手法。高次モーメントは捨象されるが，簡便かつ高速に適用可能。
   - LaplacesDemon などのソフトウェアでも標準的に実装されている (LaplacesDemon - Wikipedia)。
3. **変分ベイズ（Variational Inference; VI）**
   - 事後分布をパラメトリックな簡易分布族 $q(\theta; \phi)$ で近似し，KLダイバージェンスを最小化する最適化問題として解く。
   - 平均場近似，α-divergence 最小化，Amortized VB など多様な拡張がある ([PDF] Bayesian inference for latent variable models)。
4. **期待値伝播（Expectation Propagation; EP）**
   - 近似ファクタを逐次更新し，各因子が除かれた「残差分布」を moment-matching によりガウスで再近似する手法。VI より精度良く，ラプラス近似より堅牢とされる ([PDF] Bayesian inference for latent variable models)。
5. **マルコフ連鎖モンテカルロ（MCMC）**
   - 事後分布をターゲットとするマルコフ連鎖を構築しサンプルを得る手法。
   - 代表的アルゴリズムに Gibbs sampling，Metropolis–Hastings，Hamiltonian Monte Carlo（HMC／NUTS）などがある (LaplacesDemon - Wikipedia)。

神経回路における不確実性の表現

ここまでは最尤推定やMAP推定などにより，パラメータ(神経活動，シナプス結合)の点推定を行ってきた．**不確実性(uncertainty)** を神経回路で表現する方法として主に2つの符号化方法，**サンプリングに基づく符号化(sampling-based coding; SBC or neural sampling model)** および**確率的集団符号化(probabilistic population coding; PPC)** が提案されている．SBCは神経活動が元の確率分布のサンプルを表現しており，時間的に多数の活動を集めることで元の分布の情報が得られるというモデルである．PPCは神経細胞集団により，確率分布を表現するというモデルである．

- (Walker et al., 2022)がまとめ．
- (Fiser et al., 2010)の比較表を入れる．
- 神経活動の変動性 (neural variability)
- 自発活動が事前分布であるという説 {cite:p} `Fiser2010-kw` , {cite:p} `Berkes2011-it` .
- {cite:p} `Hoyer2002-ci`
- {cite:p} `Sanborn2016-en`

神経細胞あるいは細胞集団が確率分布を表現するにはどうすればよいだろうか．神経細胞の活動がある変数を表現していると仮定しよう．単一の細胞の瞬時的な活動がある変数の点推定に対応していると考えれば，単一の細胞の多数の活動あるいは多数の細胞の瞬時的な活動により分布は表現できると考えられる (Fig.2)．

Fig. 2. 神経活動による確率分布表現の2種類の方法．(Fiser, Berkes, Orbán, & Lengyel, 2010)より引用．(a)多数の細胞の瞬時的な活動により分布を表現する符号化 (e.g. probabilistic population codes; PPCs)．(b)単一の細胞の多数の活動により分布を表現する符号化 (e.g. neural sampling codes; NSCs)．Table1は両者の比較．著者らはSampling-based codeの方が優れていると考えている．

多数の細胞の瞬時的な活動により分布を表現する符号化としては**probabilistic population codes** (Ma, Beck, Latham, & Pouget, 2006)や**distributional TD learning** (Dabney et al., 2020; Lowet, Zheng, Matias, Drugowitsch, & Uchida, 2020)などが該当する．一方で単一の細胞の多数の活動により分布を表現する符号化は**サンプリングに基づいた符号化 (sampling-based coding)** あるいは**神経サンプリング (neural sampling)** と呼ぶ．神経サンプリングの基盤となる現象は**神経活動の変動性 (neural variability)** である．これは感覚を処理する皮質領野（例えば視覚野）において同じ入力であっても神経細胞の活動が時間や試行に応じて変動する現象のことである (Stein, Gossen, & Jones, 2005)．これが単なるノイズなのか機能があるのかに関しては様々な説が提案されているが，神経活動の変動性によりMCMCが行われているという仮説は(Hoyer & Hyvärinen, 2002)において（自分の知る限り）初めて提案された．(Sanborn & Chater, 2016)は”Bayesian Brains without Probabilities”というキャッチーな題だが，MCMCとBayesian Brainの勉強にはなる．

ここで外界の状態をx, それによって生まれた感覚刺激をy, 脳内の神経結合をWとしよう．**事前分布 (prior)** を$p(x|W)$とし，**尤度 (likelihood)** を$p(y|x,\ W)$とすると，**事後分布 (posterior)**は

$$p\left(x|y\right)=\frac{p\left(y|x,\ W\right)p(x|W)}{p(y|W)} \tag{73}$$

しかし，ここでの問題は次の2点である．すなわち，

1. 神経回路で確率分布を如何にして表現するか．
2. 規格化定数 $Z=p\left(y|W\right)=\int p\left(y|x,\ W\right)p\left(x|W\right)\ dx$をどう計算するか．

- Neural Sampling Codes
- Probabilistic Population Coding
- Distributed distributional code
  RS Zemel, P Dayan, and A Pouget. Probabilistic interpretation of population codes. Neural Computation, 10(2):403–430, 1998. [8] MSahani and P Dayan. Doubly distributional population codes: Simultaneous representation of uncertainty and multiplicity. Neural Computation, 15(10):2255–2279, 2003.

神経サンプリング

サンプリングに基づく符号化(sampling-based coding; SBC or neural sampling model)をガウス尺度混合モデルを例にとり実装する．

ガウス尺度混合モデル

ガウス尺度混合 (Gaussian scale mixture; GSM) モデルは確率的生成モデルの一種である {cite:p} `Wainwright1999-cl` {cite:p} `Orban2016-tm` ． GSMモデルでは入力を次式で予測する：

$$\text{入力}=z\left(\sum\text{神経活動}\times\text{基底}\right)+\text{ノイズ} \tag{74}$$

前節までのスパース符号化モデル等と同様に，入力が基底の線形和で表されるとしている．ただし，尺度(scale)パラメータzが基底の線形和に乗じられている点が異なる．\footnote{コードは {cite:p} `Orban2016-tm` https://github.com/gergoorban/sampling_in_gsm, および {cite:p} `Echeveste2020-sh` https://bitbucket.org/RSE_1987/ssn_inference_numerical_experiments/src/master/を参考に作成した．}

事前分布

$\mathbf{x}\in\mathbb{R}^{N_x}$, $\mathbf{A}\in\mathbb{R}^{N_x\times N_y}$, $\mathbf{y}\in\mathbb{R}^{N_y}$, $\mathbf{z}\in\mathbb{R}$とする．

$$p\left(\mathbf{x}\mid\mathbf{y},z\right)=\mathcal{N}\left(z\mathbf{A}\mathbf{y},\sigma_{\mathbf{x}}^{2}\mathbf{I}\right) \tag{75}$$

事前分布を

$$p\left(\mathbf{y}\right) = \mathcal{N}\left(\mathbf{0}, \mathbf{C}\right) \tag{76}$$
$$p\left(z\right) = \Gamma(k, \vartheta) \tag{77}$$

とする．$\Gamma(k, \vartheta)$はガンマ分布であり，kは形状(shape)パラメータ，ϑは尺度(scale)パラメータである．$p\left(\mathbf{y}\right)$は$\mathbf{y}$の事前分布であり，刺激がない場合の自発活動の分布を表していると仮定する．

分散共分散行列$\mathbf{C}$の作成

$\mathbf{C}$はyの事前分布の分散共分散行列である．{cite:p} `Orban2016-tm` では自然画像を用いて作成しているが，ここでは簡単のため$\mathbf{A}$と同様に{cite:p} `Echeveste2020-sh` に従って作成する．前項で作成した通り，$\mathbf{A}$の各基底には周期性があるため，類似した基底を持つニューロン同士は類似した出力をすると考えられる．Echevesteらは$\theta \in [-\pi/2, \pi/2)$の範囲においてFourier基底を複数作成し，そのグラム行列(Gram matrix)を係数倍したものを$\mathbf{C}$と設定している．ここではガウス過程(Gaussian process)モデルとの類似性から，周期カーネル(periodic kernel)

$$K(\theta, \theta') = \exp\left[\phi_1 \cos\left(\frac{|\theta - \theta'|}{\phi_2}\right)\right] \tag{78}$$

を用いる．ここでは$|\theta - \theta'| = m\pi \ (m = 0, 1, \ldots)$の際に類似度が最大になればよいので，$\phi_2 = 0.5$とする．これが正定値行列になるように単位行列の係数倍$\epsilon\mathbf{I}$を加算し，スケーリングした上で， `Symmetric(C)` や `Matrix(Hermitian(C))` により実対象行列としたものを$\mathbf{C}$とする．$\mathbf{C}$を正定値行列にする理由はJuliaの `MvNormal` がCholesky分解を用いて多変量正規分布の乱数を生成するためである． 事前に `cholesky(C)` が実行できるか確認するのもよい．

事後分布の計算

事後分布はzと$\mathbf{y}$のそれぞれについて次のように求められる．

$$p(z \mid \mathbf{x}) \propto p(z)\mathcal{N}\left(0, z^2\mathbf{A}\mathbf{C}\mathbf{A}^\top + \sigma_x^2\mathbf{I}\right) \tag{79}$$
$$p(\mathbf{y} \mid z, \mathbf{x}) = \mathcal{N}\left(\mu(z, \mathbf{x}), \Sigma(z)\right) \tag{80}$$

ただし，

$$\Sigma(z) = \left(\mathbf{C}^{-1} + \frac{z^2}{\sigma_x^2}\mathbf{A}^\top\mathbf{A}\right)^{-1} \tag{81}$$
$$\mu(z, \mathbf{x}) = \frac{z}{\sigma_x^2}\Sigma(z)\mathbf{A}^\top\mathbf{x} \tag{82}$$

である．

最終的な予測においてzの事後分布は必要でないため，$p(\mathbf{y} \mid z, \mathbf{x})$から$z$を消去することを考えよう．厳密に行う場合，次式のように周辺化(marginalization)により，zを（積分）消去する必要がある．

$$p(\mathbf{y} \mid \mathbf{x}) = \int dz\, p(z \mid \mathbf{x}) \cdot p(\mathbf{y} \mid z, \mathbf{x}) \tag{83}$$

周辺化においては，まずzのMAP推定（最大事後確率推定）値 z_{MAP}を求める．

$$z_{\mathrm{MAP}} = \underset{z}{\mathrm{argmax}} p(z \mid \mathbf{x}) \tag{84}$$

次にz_{MAP}の周辺で$p(z \mid \mathbf{x})$を積分し，積分値が一定の閾値を超えるzの範囲を求め，この範囲でzを積分消去してやればよい．しかし，zは単一のスカラー値であり，この手法で推定するのは煩雑であるために近似手法が{cite:p} `Echeveste2017-wu` において提案されている．Echevesteらは第一の近似として，zの分布をz_{MAP}でのデルタ関数に置き換える，すなわち，$p(z \mid \mathbf{x}) \simeq \delta(z - z_{\mathrm{MAP}})$とすることを提案している．この場合，$z$は定数とみなせ，$p(\mathbf{y} \mid \mathbf{x}) \simeq p(\mathbf{y} \mid \mathbf{x}, z = z_{\mathrm{MAP}})$となる．第二の近似として，$z_{\mathrm{MAP}}$を真のコントラスト$z^*$で置き換えることが提案されている．GSMへの入力$\mathbf{x}$は元の画像を$\tilde{\mathbf{x}}$とすると，$\mathbf{x} = z^* \tilde{\mathbf{x}}$としてスケーリングされる．この入力の前処理の際に用いるz^*を用いてしまおうということである．この場合，$p(\mathbf{y} \mid \mathbf{x}) \simeq p(\mathbf{y} \mid \mathbf{x}, z = z^*)$となる．しかし，入力を任意の画像とする場合，$z^*$は未知である．簡便さと精度のバランスを取り，ここでは第一の近似，$z = z_{\mathrm{MAP}}$とする手法を用いることにする．

興奮性・抑制性神経回路によるサンプリング

前節で実装したMCMCを**興奮性・抑制性神経回路 (excitatory-inhibitory (E-I) network)** で実装する．HMCとLMCの両方を神経回路で実装する．ハミルトニアンを用いる場合，一般化座標$\mathbf{q}$を興奮性神経細胞の活動$\mathbf{u}$, 一般化運動量$\mathbf{p}$を抑制性神経細胞の活動$\mathbf{v}$に対応させる．$\mathbf{u},\ \mathbf{v}$は同じ次元のベクトルとする．$\mathbf{u}, \mathbf{v}$の時間発展はハミルトニアン$\mathcal{H}$を導入して

$$\tau \frac{d\mathbf{u}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{v}}, \quad \tau \frac{d\mathbf{v}}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{u}} \tag{85}$$

と書ける．一般的には$\mathcal{H}(\mathbf{u}, \mathbf{v}) = E\left(\mathbf{u}\right) + \frac{1}{2}\mathbf{v}^\top \mathbf{v}$であり，$p\left(\mathbf{u},\ \mathbf{v}\right) \propto \exp(-\mathcal{H}(\mathbf{u}, \mathbf{v}))$である．力学的エネルギーを保つ運動は，対数同時分布における等値線上の運動と同じである．

\citep{Aitchison2016-xu}では

$$\mathcal{H}(\mathbf{u}, \mathbf{v}) = \log p\left(\mathbf{u}, \mathbf{v}\right) + \mathrm{Const.} = \log p\left(\mathbf{v}|\mathbf{u}\right) + \log p\left(\mathbf{u}\right) + \mathrm{Const.} \tag{86}$$

とし，$p\left(\mathbf{v}|\mathbf{u}\right) = \mathcal{N}\left(\mathbf{v}; \mathbf{B}\mathbf{u},\ \mathbf{M}^{-1}\right),\ p\left(\mathbf{u}\right) = \mathcal{N}\left(\mathbf{0},\ \mathbf{C}^{-1}\right)$としている．この場合，

$$\frac{d\mathbf{u}}{dt} = \frac{1}{\tau}\frac{\partial \mathcal{H}}{\partial \mathbf{v}} = \frac{1}{\tau}\frac{\partial \log p\left(\mathbf{u},\ \mathbf{v}\right)}{\partial \mathbf{v}} = \frac{1}{\tau}\frac{\partial \log p\left(\mathbf{v}|\mathbf{u}\right)}{\partial \mathbf{v}} \tag{87}$$

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\tau}\frac{\partial \mathcal{H}}{\partial \mathbf{u}} = -\frac{1}{\tau}\frac{\partial \log p\left(\mathbf{u},\ \mathbf{v}\right)}{\partial \mathbf{u}} = -\frac{1}{\tau}\frac{\partial \log p\left(\mathbf{v}|\mathbf{u}\right)}{\partial \mathbf{u}} - \frac{1}{\tau}\frac{\partial \log p\left(\mathbf{u}\right)}{\partial \mathbf{u}} \tag{88}$$

となる．このままでは等値線上を運動することになるので，Langevinダイナミクスを付け加える．

$$\frac{d\mathbf{u}}{dt} = \frac{1}{\tau}\frac{\partial \log p\left(\mathbf{v}|\mathbf{u}\right)}{\partial \mathbf{v}} + \frac{1}{\tau_L}\frac{\partial \log p\left(\mathbf{u},\ \mathbf{v}\right)}{\partial \mathbf{u}} + \sqrt{\frac{2}{\tau_L}}\,d\eta \tag{89}$$

$$= \frac{1}{\tau}\frac{\partial \log p\left(\mathbf{v}|\mathbf{u}\right)}{\partial \mathbf{v}} + \frac{1}{\tau_L}\frac{\partial \log p\left(\mathbf{v}|\mathbf{u}\right)}{\partial \mathbf{u}} + \frac{1}{\tau_L}\frac{\partial \log p\left(\mathbf{u}\right)}{\partial \mathbf{u}} + \sqrt{\frac{2}{\tau_L}}\,d\eta \tag{90}$$

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\tau}\frac{\partial \log p\left(\mathbf{v}|\mathbf{u}\right)}{\partial \mathbf{u}} - \frac{1}{\tau}\frac{\partial \log p\left(\mathbf{u}\right)}{\partial \mathbf{u}} + \frac{1}{\tau_L}\frac{\partial \log p\left(\mathbf{u},\mathbf{v}\right)}{\partial \mathbf{v}} + \sqrt{\frac{2}{\tau_L}}\,d\eta \tag{91}$$

$$= -\frac{1}{\tau}\frac{\partial \log p\left(\mathbf{v}|\mathbf{u}\right)}{\partial \mathbf{u}} + \frac{1}{\tau_L}\frac{\partial \log p\left(\mathbf{v}|\mathbf{u}\right)}{\partial \mathbf{v}} - \frac{1}{\tau}\frac{\partial \log p\left(\mathbf{u}\right)}{\partial \mathbf{u}} + \sqrt{\frac{2}{\tau_L}}\,d\eta \tag{92}$$

となる．それぞれの項は

$$\frac{\partial \log p\left(\mathbf{v}|\mathbf{u}\right)}{\partial \mathbf{v}} = \mathbf{B}^{\top}\mathbf{M}\left(\mathbf{B}\mathbf{u} - \mathbf{v}\right) \tag{93}$$

$$\frac{\partial \log p\left(\mathbf{v}|\mathbf{u}\right)}{\partial \mathbf{u}} = -\mathbf{M}\left(\mathbf{B}\mathbf{u} - \mathbf{v}\right) \tag{94}$$

$$\frac{\partial \log p\left(\mathbf{u}\right)}{\partial \mathbf{u}} = -\mathbf{C}\mathbf{u} \tag{95}$$

であるので，

$$\frac{d\mathbf{u}}{dt} = \frac{1}{\tau}\mathbf{B}^{\top}\mathbf{M}\left(\mathbf{B}\mathbf{u} - \mathbf{v}\right) - \frac{1}{\tau_L}\mathbf{M}\left(\mathbf{B}\mathbf{u} - \mathbf{v}\right) - \frac{1}{\tau_L}\mathbf{C}\mathbf{u} + \sqrt{\frac{2}{\tau_L}}\,d\eta \tag{96}$$

$$\frac{d\mathbf{v}}{dt} = \frac{1}{\tau}\mathbf{M}\left(\mathbf{B}\mathbf{u} - \mathbf{v}\right) + \frac{1}{\tau_L}\mathbf{B}^{\top}\mathbf{M}\left(\mathbf{B}\mathbf{u} - \mathbf{v}\right) + \frac{1}{\tau}\mathbf{C}\mathbf{u} + \sqrt{\frac{2}{\tau_L}}\,d\eta \tag{97}$$

となる．$\mathbf{B} = \mathbf{I}$ とすると，

$$\frac{d\mathbf{u}}{dt} = \frac{1}{\tau}\mathbf{M}\left(\mathbf{u}-\mathbf{v}\right) - \frac{1}{\tau_L}\mathbf{M}\left(\mathbf{u}-\mathbf{v}\right) - \frac{1}{\tau_L}\mathbf{C}\mathbf{u} + \sqrt{\frac{2}{\tau_L}}\,d\eta \tag{98}$$

$$= \left[\left(\frac{1}{\tau}-\frac{1}{\tau_L}\right)\mathbf{M} - \frac{1}{\tau_L}\mathbf{C}\right]\mathbf{u} - \left(\frac{1}{\tau}-\frac{1}{\tau_L}\right)\mathbf{M}\mathbf{v} + \sqrt{\frac{2}{\tau_L}}\,d\eta \tag{99}$$

$$\frac{d\mathbf{v}}{dt} = \frac{1}{\tau}\mathbf{M}\left(\mathbf{u}-\mathbf{v}\right) + \frac{1}{\tau_L}\mathbf{M}\left(\mathbf{u}-\mathbf{v}\right) + \frac{1}{\tau}\mathbf{C}\mathbf{u} + \sqrt{\frac{2}{\tau_L}}\,d\eta \tag{100}$$

$$= \left[\left(\frac{1}{\tau}+\frac{1}{\tau_L}\right)\mathbf{M} + \frac{1}{\tau_L}\mathbf{C}\right]\mathbf{u} - \left(\frac{1}{\tau}+\frac{1}{\tau_L}\right)\mathbf{M}\mathbf{v} + \sqrt{\frac{2}{\tau_L}}\,d\eta \tag{101}$$

となり，$\mathbf{u}$, $\mathbf{v}$と定行列およびノイズに依存してサンプリングダイナミクスを記述できる．長々と式変形を書いたが，重要なのは**興奮性・抑制性という2種類の細胞群の相互作用により生み出された振動を用いてサンプリングにおける自己相関を下げることができる**という点である．

簡単のため，前項で用いた入力刺激のうち，最もzが大きいサンプルのみを使用する．

Hamiltonianネットワークは自己相関を振動により低下させることで，効率の良いサンプリングを実現している．ToDo: 普通にMCMCやる場合も自己相関は確認したほうがいいという話をどこかに書く．

推定された事後分布を特定の神経細胞のペアについて確認する．

Hamiltonianネットワークの方が安定して事後分布を推定することができている．ToDo: 以下の記述．ここでは重みを設定したが，{cite:p} `Echeveste2020-sh` ではRNNにBPTTで重みを学習させている．動的な入力に対するサンプリング {cite:p} `Berkes2011-xj` ．burn-inがなくなり効率良くサンプリングできる．

Spikingニューラルネットワークにおけるサンプリング

前項で挙げた例は発火率モデルであったが，SNNにおいてサンプリングを実行する機構自体は考案されている．ToDo: 以下の記述．{cite:p} `Buesing2011-dm` {cite:p} `Masset2022-wh` {cite:p} `Zhang2022-bl`

シナプスサンプリング

ここまでシナプス結合強度は変化せず，神経活動の変動によりサンプリングを行うというモデルについて考えてきた．一方で，シナプス結合強度自体が短時間で変動することによりベイズ推論を実行するというモデルがあり，**シナプスサンプリング(synaptic sampling)** と呼ばれる．ToDo: 以下の記述．{cite:p} `Kappel2015-kq` {cite:p} `Aitchison2021-wo`

確率的集団符号化

確率的集団符号化 (probabilistic population coding)

Distributional Population Coding or distributed distributional codes (DDCs)

ポアソン分布

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!} \tag{102}$$

より，

$$p(y \mid \mathbf{x}) \propto \prod_i \frac{e^{-f_i(y)} f_i(y)^{x_i}}{x_i!} p(y) \tag{103}$$