

双方向予測符号化の スパイキングニューラルネットワークへの 実装による計算コストの削減

創域理工学部 情報計算科学科 桂田研究室 4年
学籍番号 : 6322045
砂川恵太郎

2026 年 2 月 8 日

1 要旨

2 背景

近年、深層学習を筆頭とする機械学習技術は目覚ましい進歩を遂げ、特に大規模言語モデル（LLM）は、自然言語処理の分野において人間を凌駕する性能を示すまでになった。しかしその一方で、その成功は膨大な計算資源とデータ量を前提としており、生物の脳が持つ圧倒的なエネルギー効率や柔軟性とは未だ大きな乖離がある。また、現在の LLM の多くはテキストという単一のモダリティに特化しており、我々が日常的に経験するような、複数の感覚情報を統合して世界を認識する能力については疑問を感じる。

この課題を克服し、汎用的な知能を獲得するためには、マルチモーダルな情報を扱え、かつ生物学的な妥当性を備えた新しい計算モデルの探求が不可欠である。そこで本稿では、数ある感覚の中でも特に外界理解の根幹をなす「視覚」情報処理に着目し、脳の情報処理原理に基づいた新たなネットワークモデルを提案する。

機械学数の根幹をなすニューロネットワークのモデルについて、ここでは生物学的妥当性の高いものとしてスパイクニューラルネットワークを採用する。スパイクニューラルネットワークのニューロンモデルは、人の脳の神経活動をより忠実に再現したモデルであり、ニューロン間の通信を $\{0, 1\}$ のデジタル値で行うことが最大の特徴である。これにより通信の必要がないときは情報のやり取りは行われず、通信が必要なときは 1 という 1 ビットの情報だけが送られるため、通信にかかる計算コストを大幅に削減することができる。しかし通信をデジタル値で行う以上、その微分不可能性から強力な学習則である誤差逆伝播法をそのまま用いることができない。また、生物学的には、誤差逆伝播法のようにネットワーク全体で誤差を共有するというような大域的な学習則が脳内で用いられているとは考えられにくく、その点でも誤差逆伝播法はミスマッチであるといえる。

ここでスパイクニューラルネットワークの有効な学習則となり得るのが、予測符号化理論である。予測符号化理論は、外部からの感覚入力に対して予測を立て、予測と感覚入力の差、すなわち予測誤差を最小化することで入力に対する良いモデルを形作ることを骨子としている。この誤差はモデルを階層構造とするならば、直近の層としかやり取りされない情報であり、ローカルな学習則であるといえる。

この 2 つの理論を用いて、生物学的に妥当なネットワークを提案していくのが、本研究のメインテーマとなる。

3 関連研究

本研究は、主に「Difference Predictive Coding (DiffPC)」と「Bidirectional Predictive Coding (bPC)」という 2 つの関連研究を基盤としている。

3.1 Difference Predictive Coding

「Difference Predictive Coding」は、予測符号化理論をスパイクニューラルネットワーク上に実装した研究である。特徴として、セクション 2 ではスパイクニューラルネットワークは $\{0, 1\}$ のデジタル値をスパイクとすると述べたが、このモデルでは $\{-1, 0, 1\}$ の 3 値のスパイクに拡張している。これにより、スパイクに誤差が正か負かの情報を付与することができ、値をどの方向に修正すればよいか判断することができる。また、発生させるスパイクの量はやり取りの対象となっている情報の、時刻 t と $t+1$ の中で変化した分だけ

を通信するのに限るため、通信する際の情報量を削減できる。

3.1.1 スパイキングニューラルネットワーク

DiffPC に採用されているニューロンモデルは、時刻 t に対する次の差分方程式で表される。

$$V_i(t+1) = V_i(t) - T_\theta s_i(V_i(t)) + \sum_j w_{ij} s_j(t), \quad V_i(0) = b_i \quad (1)$$

ここで、 $V_i(\cdot)$ はニューロン i の膜電位、 T_θ は膜電位の閾値、 $s_i(\cdot)$ はスパイクの発生関数、 w_{ij} はニューロン i と j を結ぶ重みであり b_i はバイアスである。スパイクの発生関数 $s_i(\cdot)$ は、次のように定義される。

$$s_i(V_i(t)) := \begin{cases} 1 & \text{if } V_i(t) \geq T_\theta \\ -1 & \text{if } V_i(t) \leq -T_\theta \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

閾値をどちらの方向に膜電位が超えたかによって、スパイクの正負が決定するのがわかる。また、式 (1) と式 (2) を合わせてみると、スパイク発火後はそのスパイクの正負にかかわらず、膜電位は入力を除いた部分が 0 にリセットされるのが確認できる。

3.1.2 予測符号化

ニューラルネットワークに予測符号化を導入するときの動作として、各層が次の層の活動の予測をするというのがよく用いられる。順伝播として予測を流し、逆伝播として各層の活動と予測の差である誤差を流すサイクルの中で、ネットワーク全体の誤差の最小化を通して学習を実行する。図 1 はニューラルネットワーク上で

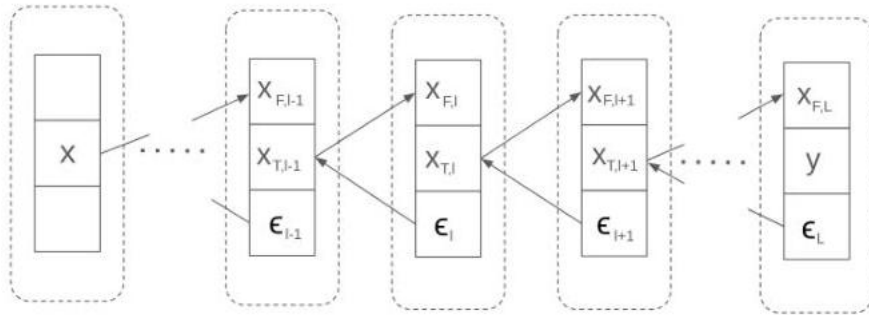


図 1 ニューラルネットワーク上での予測符号化

の予測符号化を図式化したものである。入力 x と出力 y に対して、各隠れ層 l は現在の状態を示す $x_{T,l}$ (T はターゲットを示す) と下位層からの予測を示す $x_{F,l}$ (F はフォワードを示す)、 $x_{T,l}$ と $x_{F,l}$ の差を示す ϵ_l を保持している。層間の矢印は、始点の情報を終点の変数に伝達することを示しており、順伝播として予測を渡し逆伝播として誤差を渡す。

ここからは、予測符号化における予測誤差の最小化を、エネルギー関数を最小化するという観点で記述していく。第 l 層に対する予測 $x_{F,l}$ は、第 $l-1$ 層の活動 $x_{T,l-1}$ を用いて次のように表される。

$$x_{F,l} = \mathbf{W}_l \phi(x_{T,l-1}) \quad (3)$$

ただし \mathbf{W}_l は第 $l-1$ 層と第 l 層を結合する重みであり、 $\phi(\cdot)$ は活性化関数である。これより第 l 層における

活動と予測の誤差である予測誤差 ϵ_l は、次のように定められる。

$$\epsilon_l = \mathbf{x}_{T,l} - \mathbf{x}_{F,l} \quad (4)$$

全ての層の予測誤差の2乗和をとり、それをネットワークのエネルギー関数 E として最小化を考える。

$$E = \sum_{l=1}^L \|\epsilon_l\|_2^2 = \sum_{l=1}^L (\mathbf{x}_{T,l} - \mathbf{x}_{F,l})^2 \quad (5)$$

エネルギー関数の最小化は、推論活動としてのパラメータ $\mathbf{x}_{T,l}$ と、学習則としてのパラメータ \mathbf{W}_l に対して行われる。 $\mathbf{x}_{T,l}$ の更新則は次のように表される。

$$\mathbf{x}_{T,l}^{(t+1)} = \mathbf{x}_{T,l}^{(t)} - \gamma \frac{\partial E}{\partial \mathbf{x}_{T,l}} = \begin{cases} \mathbf{x}_{T,l}^{(t)} - \gamma (\epsilon_l - \mathbf{W}_{l+1}^\top \epsilon_{l+1} \odot \phi'(\mathbf{x}_{T,l})) & \text{for } l < L \\ \mathbf{x}_{T,l}^{(t)} - \gamma \epsilon_l & \text{for } l = L \end{cases} \quad (6)$$

ただし γ は推論活動における学習率であり、 \odot はアダマール積である。続いて重み \mathbf{W}_l の更新則は次のように表される。

$$\mathbf{W}_l^{(t+1)} = \mathbf{W}_l^{(t)} - \alpha \frac{\partial E}{\partial \mathbf{W}_l} = \mathbf{W}_l^{(t)} + \alpha \epsilon_l \phi(\mathbf{x}_{T,l-1})^\top \quad (7)$$

ただし α は学習則における学習率である。これらの更新則には第 $l-1$ 層と第 l 層の誤差と活動のみ用いられているため、ローカルなルールで推論と学習を進めることができる。

3.1.3 アルゴリズム

セクション 3.1.2 の予測符号化フレームワークをスパイクニューラルネットワークに実装するためには、ニューロン活動 $x_{T,l}$ に用いられている浮動小数点演算と情報通信をスパイクに変換する必要がある。このアルゴリズムではこれらの情報は $\{-1, 0, 1\}$ の3値に変換される。

3.2 双方向予測符号化 (bPC)

予測符号化とは、脳が高効率な情報処理を実現するための一つの理論であり、「自身の予測と実際の入力との差分（予測誤差）を最小化する」という原理に基づいている。予測符号化の実装においてはトップダウンもしくはボトムアップどちらかの単方向予測モデルが多いが、bPC は双方向で予測を行うモデルである。

生成と識別の両立

- bPC は、トップダウン（上位層から下位層へ）の生成的予測と、ボトムアップ（下位層から上位層へ）の識別予測を単一のエネルギー関数で定式化し、両方の予測誤差を同時に最小化する。
- これにより、教師あり学習における高い分類性能と、教師なし学習における優れた表現学習能力を両立させる。

3.2.1 理論

エネルギー関数

bPC のエネルギー関数 E は以下で定義される。

$$E(x, W, V) = \sum_{l=1}^{L-1} \frac{\alpha_{gen}}{2} \|\mathbf{x}_l - \mathbf{W}_{l+1} f(\mathbf{x}_{l+1})\|_2^2 + \sum_{l=2}^L \frac{\alpha_{disc}}{2} \|\mathbf{x}_l - \mathbf{V}_{l-1} f(\mathbf{x}_{l-1})\|_2^2 \quad (8)$$

ここで、 x_l は第 l 層のニューロン活動、 W_l と V_l はそれぞれトップダウンとボトムアップの重み行列、 f は活性化関数、 α_{gen} と α_{disc} は生成と識別のバランスを調整する係数である。

学習則

ニューロン活動 x_l および重み W, V は、勾配降下法によりエネルギー E を最小化するように更新される。

■ニューロン活動の更新:

$$\frac{dx_l}{dt} = -\nabla_x E = -\epsilon_l^{gen} - \epsilon_l^{disc} + f'(x_l) \odot (W_l^\top \epsilon_{l-1}^{gen} + V_l^\top \epsilon_{l+1}^{disc}) \quad (9)$$

ここで、 $\epsilon_l^{gen} := \alpha_{gen}(x_l - W_{l+1}f(x_{l+1}))$ と $\epsilon_l^{disc} := \alpha_{disc}(x_l - V_{l-1}f(x_{l-1}))$ はそれぞれトップダウン予測誤差とボトムアップ予測誤差である。

■重みの更新:

$$\Delta W_l \propto \epsilon_{l-1}^{gen} f(x_l)^\top, \quad \Delta V_l \propto \epsilon_{l+1}^{disc} f(x_l)^\top \quad (10)$$

神経実装

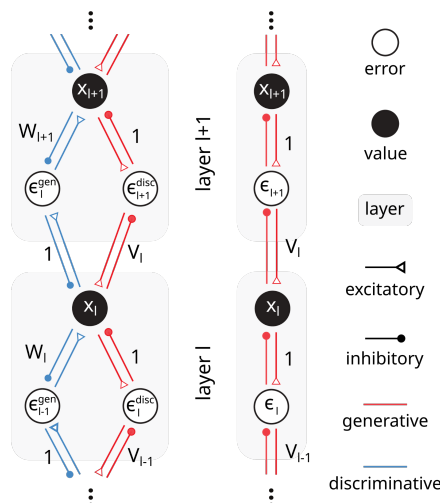


図2 bPC の神経実装

図2はbPCモデルの神経実装である。左のモデルがbPCであり、中央のモデルは識別過程のみの予測符号化モデルである。bPCモデルの各ニューロンの添え字に注目すると、価値ニューロンと同じ添え字のエラーニューロンは重み1のシナプスで結合されているのが確認できる。これはニューロン活動の更新式から分かるように、同じ層の価値ニューロンは自らの値を同じ層のエラーニューロンに送る必要があり、逆にエラーニューロンは、その結果となる予測誤差を価値ニューロンにフィードバックする必要があるからである。異なる層のニューロン同士では、その予測に重みを乗算して渡す必要があるので、シナプス結合の強さは W と V になる。

柔軟な学習

bPCは、教師あり学習と教師なし学習の両方で学習できる。いずれの場合も、中間層（層2から $L-1$ ）のニューロンは入力にクランプされず、式(9)に示すニューラルダイナミクスに従って進化する。教師あり学習では、第1層 x_1 は入力であるデータそのものに固定され、最上層 x_L はターゲットラベルに固定される。教師なし学習では、 x_L は固定されず、bPCは入力の圧縮表現を学習する。 x_L 内のニューロンのサブセットのみがラベル情報に固定され、他のニューロンは固定されない混合学習も可能である。この設定では、モデルはラベルを推論し、関連する圧縮表現を学習することができる。

3.2.2 実験

教師あり分類と生成

この実験では、bPCモデルが単一のモデルで識別タスク（画像分類）と生成タスク（クラス平均画像の生成）を同時に実行できることを示します。MNISTとFashion-MNISTデータセットを用いて、他の単方向予測符号化モデル（discPC（識別的予測符号化）、genPC（生成的予測符号化）、hybridPC（双方向予測符号化、ただしbPCとは異なる））や、それらに対応するバックプロパゲーションベースのモデル（BP）と比較評価を行った。

実験結果

表 1 と図 3 に結果を示す．分類性能は正解率（Accuracy, %）で，生成性能は元画像のクラス平均と生成画像の間の RMSE（Root Mean Squared Error）で評価している．結果として，bPC は分類精度において純

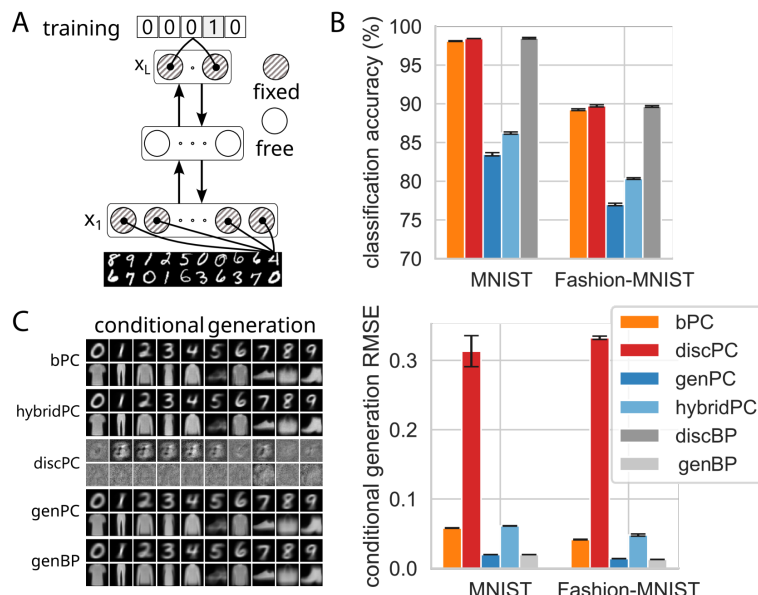


図 3 bPC は，MNIST と Fashion-MNIST を正確に分類し，クラス平均画像を生成する．A：トレーニングセットアップ．モデルは， x_1 を画像に固定し， x_L をラベルに固定してトレーニングされる．B：モデルの分類精度．C：クラスラベルを条件として生成された画像の例（左）と，生成された画像と各クラスの平均画像間の RMSE（右）．

表 1 教師あり学習における分類精度と画像生成 RMSE

モデル	分類正解率 (Acc %) ↑		生成 RMSE ↓	
	MNIST	Fashion-MNIST	MNIST	Fashion-MNIST
bPC	98.10 ± 0.05	89.24 ± 0.12	0.0581 ± 0.0004	0.0415 ± 0.0005
hybridPC	86.22 ± 0.15	80.34 ± 0.11	0.0612 ± 0.0003	0.0480 ± 0.0016
genPC	83.48 ± 0.21	77.00 ± 0.16	0.0198 ± 0.0001	0.0140 ± 0.0001
discPC	98.43 ± 0.01	89.74 ± 0.14	0.3133 ± 0.0224	0.3326 ± 0.0024
BP (genBP/discBP)	98.48 ± 0.10	89.66 ± 0.11	0.0198 ± 0.0001	0.0128 ± 0.0001

粋な識別モデルである discPC や BP と同等の性能を達成し，同時に画像生成においては純粋な生成モデルである genPC や genBP に匹敵する低い RMSE を示した．これにより，bPC が識別と生成の両方のタスクを単一モデルで高いレベルで両立できることが確認された．

教師なし表現学習

このセクションでは，教師なし設定で bPC が効果的に圧縮表現を学習する能力を評価する．MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100 データセットを使用し，入力画像からの再構成誤差（RMSE）を指

標として、他のモデルと比較を行う。

実験結果

表 2 と図 4 に、各モデルが学習した表現から画像を再構成した際の RMSE を示す。図 4B および表 2 が示す

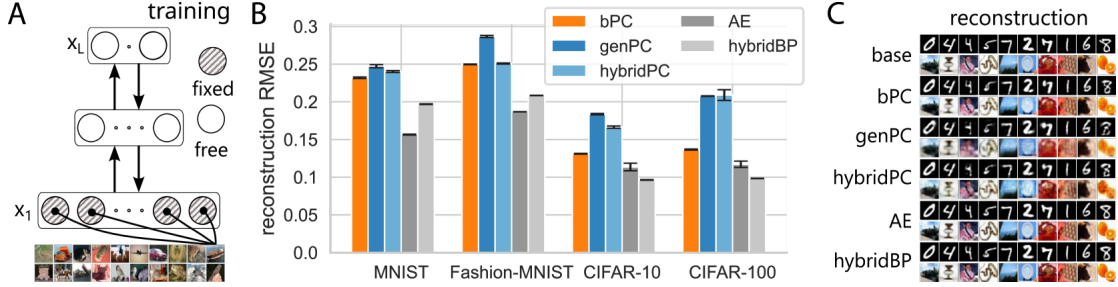


図 4 bPC は、低次元表現の学習において他の PC モデルよりも優れた性能を発揮。A：入力画像に x_1 のみが固定されたトレーニングセットアップ。B：モデルとデータセット間の再構成 RMSE。C：再構成の例。

表 2 教師なし学習で得られた表現からの画像再構成 RMSE

モデル	MNIST	Fashion-MNIST	CIFAR-10	CIFAR-100
bPC	0.2320 ± 0.0010	0.2497 ± 0.0004	0.1311 ± 0.0005	0.1366 ± 0.0007
genPC	0.2473 ± 0.0020	0.2868 ± 0.0013	0.1837 ± 0.0009	0.2077 ± 0.0003
hybridPC	0.2401 ± 0.0012	0.2508 ± 0.0007	0.1664 ± 0.0015	0.2089 ± 0.0071
AE	0.1565 ± 0.0006	0.1868 ± 0.0001	0.1135 ± 0.0050	0.1171 ± 0.0042
BP (hybridBP)	0.1969 ± 0.0004	0.2084 ± 0.0002	0.0964 ± 0.0004	0.0983 ± 0.0002

ように、bPC は全てのデータセットにおいて genPC を一貫して上回り、より複雑な CIFAR データセットでは hybridPC を大幅に上回る性能を示した。これは、bPC の双方向アーキテクチャが、ボトムアップの重みも推論ダイナミクスに積極的に関与させることで、より高品質な表現学習を可能にすることを示唆している。

3.3 Spiking Neural Predictive Coding (SpNCN)

SpNCN は、予測符号化の理論をスパイクニューラルネットワーク (SNN) 上で実現したモデルである。SNN は、生物の神経細胞 (ニューロン) の発火現象を模したネットワークであり、非同期かつイベント駆動で動作するため、エネルギー効率が非常に高いという特徴がある。

3.3.1 理論

ニューロンモデル

SpNCN では、ニューロンモデルとして漏れ積分発火 (Leaky Integrate-and-Fire, LIF) モデルが用いられる。ニューロンの膜電位 $v^l(t)$ は、入力電流 $j^l(t)$ に応じて時間変化し、以下の微分方程式で記述される。

$$\tau_m \frac{\partial v^l}{\partial t} = -\gamma_m v^l(t) + R_m j^l(t) \quad (11)$$

ただし、 τ_m は漏れの時間スケールを扱う膜時定数で、 γ_m は漏れの強さ、 R_m は膜抵抗である。膜電位が特定の閾値 v_{thr} を超えると、ニューロンは**発火（スパイク）**し、デジタル値「1」を出力する。発火後、膜電位はリセットされる。また、電圧 $v^l(t)$ はオイラー法を用いて離散化され、微小時間 Δt に対して更新される。

$$v^l(t + \Delta t) = v^l(t) + \frac{\Delta t}{\tau_m}(-\gamma_m v^l(t) + R_m j^l(t)) \quad (12)$$

予測と誤差計算

離散的なスパイク列 $s^l(t)$ は、時間定数 τ_{tr} を持つローパスフィルタによって滑らかな連続値（トレース） $z^l(t)$ に変換される。

$$z^l(t) = z^l(t) + \left(-\frac{z^l(t)}{\tau_{tr}} + s^l(t) \right) \quad (13)$$

このトレースを用いて、トップダウン予測 z_μ^l と予測誤差 $e^l(t)$ が計算される。ただし、 $W^l(t)$ はトップダウン予測シナプス結合の重み行列である。

$$z_\mu^l = W^{l+1} \cdot s^{l+1}(t) \quad (14)$$

$$e^l(t) = (z^l(t) - z_\mu^l) \quad (15)$$

学習則

入力電流 $j^l(t)$ は予測誤差 $e^l(t)$ と、下位層からの誤差フィードバック $E \cdot e^{l-1}(t)$ を用いた時間に関する微分方程式で表される。

中間層 l :

$$\tau_j \frac{\partial J^l(t)}{\partial t} = -\kappa_j J^l(t) + \phi_e(-e^l(t) + E^l \cdot e^{l-1}(t)) \quad (16)$$

最終層 L :

$$\tau_j \frac{\partial J^l(t)}{\partial t} = -\kappa_j J^l(t) + \phi_e(E^l \cdot e^{l-1}(t)) \quad (17)$$

ただし、 τ_j は電流の漏れに関する時定数であり、 κ_j は電流の漏れ、 ϕ_e は誤差信号に対する変換関数である。これらの信号の各ニューロン間での動きは、図 5 に示されている。

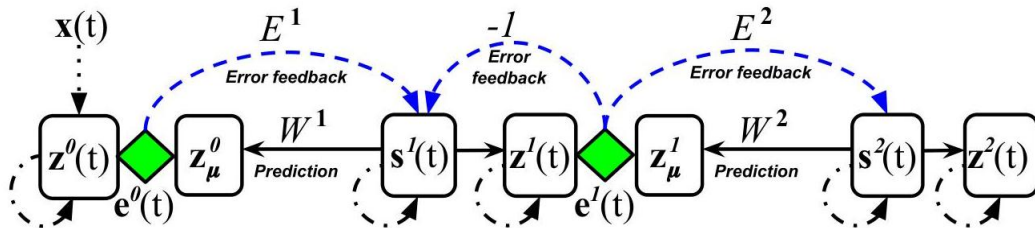


図 5 2層 SpNCN モデル

このダイナミクスの中で、電流 $j^l(t)$ はオイラー法により離散化され、 Δt に対して更新される。

中間層 l :

$$j^l(t + \Delta t) = j^l(t) + \frac{\Delta t}{\tau_j} (-\kappa_j J^l(t) + \phi_e(-e^l(t) + E^l \cdot e^{l-1}(t))) \quad (18)$$

最終層 L :

$$\mathbf{j}^l(t + \Delta t) = \mathbf{j}^l(t) + \frac{\Delta t}{\tau_j} (-\kappa_J \mathbf{J}^l(t) + \phi_e(\mathbf{E}^l \cdot \mathbf{e}^{l-1}(t))) \quad (19)$$

最後に、トップダウン予測シナプス結合の重み行列 \mathbf{W}^l と誤差フィードバック重み行列 \mathbf{E}^l は、局所的な情報のみを用いて、勾配降下法によって更新される。

$$\Delta \mathbf{W}^l = \mathbf{e}^{l-1}(t) \cdot (\mathbf{s}^l(t))^\top \quad (20)$$

$$\Delta \mathbf{E}^l = \beta (\mathbf{s}^l(t) \cdot (\mathbf{e}^{l-1}(t))^\top) \quad (21)$$

ただし、 β は \mathbf{E}^l の更新速度を制御する係数である。

3.3.2 実験

表 3は、手書き数字データセットである MNIST に対する、様々なスパイキングネットワークモデルの汎化誤差 (Generalization error) を比較したものである。汎化誤差は値が低いほど性能が良いことを示す。この

表 3 MNIST におけるスパイキングネットワークの汎化誤差 (低いほど良い)。SpNCN および実装された SNN ベースライン ("impl." とラベル付け) の性能は 10 試行で測定 (平均と標準偏差を報告)。二重線の下にあるモデルはオンラインモデルである。

モデル	前処理	Type	Performance
Dendritic Neurons	はい	Rate-based	9.7%
Spiking RBM	いいえ	Rate-based	11.0%
Spiking RBM	はい	Rate-based	5.9%
Spiking CNN, BP	いいえ	Rate-based	0.9%
Spiking RBM	はい	Rate-based	7.4%
Spiking RBM	はい	Spike-Based	8.1%
Spiking CNN	はい	Spike-Based	8.7%
2-Layer SNN	はい	Spike-Based	3.5%
ML H-SNN	はい	Spike-Based	8.4%
2-Layer SNN	いいえ	Spike-Based	6.5%
2-Layer SNN	いいえ	Spike-Based	5.0%
syn-SNN (STDP)	いいえ	Rate-based	3.27%
SNN-LM	いいえ	Spike-Based	5.93%
2-Layer SNN, 3 passes	いいえ	Spike-Based	~ 17.1%
Online SNN-LM	いいえ	Spike-Based	6.61%
Online SCNN	いいえ	Spike-Based	4.76%
SNN, df-DRTP (impl.)	いいえ	Spike-Based	40.13 \pm 0.35%
SNN, df-BFA (impl.)	いいえ	Spike-Based	9.38 \pm 0.12%
SpNCN (ours)	いいえ	Spike-Based	2.47 \pm 0.16%

表の核心は、提案手法である **SpNCN** の性能を他のモデルと比較することにある。

■**SpNCN の優れた性能** 提案手法である **SpNCN** は、汎化誤差 **2.47%** を達成しており、これは表中の他の多くの SNN モデルを大幅に上回る結果である。特に注目すべきは、二重線の下に示された「オンライン学習モデル」の中での性能である。オンライン学習とは、データを一度しか見ずに学習する、より実践的で困難な設定である。SpNCN はこの設定でありながら、複数回データを見て学習するオフラインモデルに匹敵する、あるいはそれ以上の性能を示している。

■**他のオンラインモデルとの比較** 同じオンライン学習の SNN モデルである「Online SCNN」(4.76%) や「Online SNN-LM」(6.61%) と比較して、SpNCN の誤差率は約半分であり、その優位性は明らかである。また、著者らが比較対象として実装した「SNN, df-BFA」(9.38%) や「SNN, df-DRTP」(40.13%) と比較しても、提案された学習手法 (ST-LRA) がいかに効果的であるかが分かる。

■**学習条件の利点** SpNCN は「前処理」の項目が「いいえ」となっており、事前のデータ加工なしで高い性能を達成している。これは、モデルがデータから直接特徴を学習する能力が高いことを示唆しており、実用上の利点となる。

■**結論** 表 3 は、SpNCN が MNIST データセットにおいて、特にオンライン学習という制約の厳しい条件下で、既存の SNN モデルよりも優れた性能を発揮することを示した、重要な結果である。

また表 4 は、MNIST よりも複雑で多様な 3 つのデータセット (Fashion MNIST, Stanford OCR, Caltech 101) を用いて、オンラインスパイクングネットワークの汎化誤差を比較したものである。これにより、SpNCN の性能が MNIST 以外のタスクにも汎化可能かどうかを検証している。以下のモデルはすべてデータの前処理を行っていない。この表 4 は、SpNCN の汎化性能をさらに厳しく評価するものである。

表 4 Fashion MNIST, Stanford OCR, Caltech 101 データセットにおける様々なオンラインスパイクングネットワークの汎化誤差 (低いほど良い)。SpNCN および実装された SNN ベースラインの性能は 10 試行で測定 (平均と標準偏差を報告)。

Model	Type	Fashion MNIST Performance	Stanford OCR Performance	Caltech 101 Performance
ANN (BP)	Rate-based	12.98%	37.01 \pm 1.19%	44.77 \pm 1.22%
syn-SNN (STDP)	Rate-based	15.35%	—	—
SNN, df-DRTP (impl.)	Spike-Based	45.27 \pm 0.26%	82.19 \pm 0.12%	77.75 \pm 0.24%
SNN, df-BFA (impl.)	Spike-Based	25.65 \pm 0.12%	84.11 \pm 0.51%	63.68 \pm 0.42%
SpNCN (ours)	Spike-Based	15.61 \pm 0.05%	42.78 \pm 0.16%	51.67 \pm 0.53%

■**一貫した優位性** 最も重要な点は、**SpNCN** がテストされた 3 つのデータセットすべてにおいて、他のオンライン SNN ベースライン (df-DRTP, df-BFA) を一貫して大幅に上回っていることである。これにより、SpNCN のアーキテクチャと学習メカニズムが、特定のデータセットに過度に依存するものではなく、ロバストであることが示される。

■Fashion MNIST での性能 Fashion MNIST において, SpNCN の誤差 (15.61%) は, 同じく SNN である「syn-SNN (STDP)」(15.35%) に匹敵し, さらには標準的な非スパイキングニューラルネットワークである「ANN (BP)」(12.98%) に迫る性能を示している. スパイキングモデルは一般的に ANN よりも学習が難しいとされる中で, この結果は非常に注目に値する.

■Stanford OCR と Caltech 101 での性能 これら 2 つのより困難なデータセットでは, 全体的にモデルの誤差は高くなっている. しかし, その中でも SpNCN は他の SNN モデルに対して明確な性能的優位性を保っている. 例えば Caltech 101 では, SpNCN の誤差は 51.67% であるのに対し, 次点の SNN, df-BFA は 63.68% であり, 10% 以上の差がある. この結果は, 多様な特徴を持つ画像に対しても SpNCN が比較的うまく機能することを示唆している.

■結論 表 4 は, 表 3 で示された SpNCN の優れた性能が, MNIST という単一のタスクに限定されたものではなく, より多様で複雑な画像分類タスクにおいても維持されることを実証している. これは, 提案手法の汎用性と実用的な可能性を強く裏付けるものである.

4 提案手法

本研究では, bPC の持つ「生成と識別の両立」という強力な学習能力と, SpNCN の持つ「高いエネルギー効率」を統合した新しいモデル「Bidirectional SpNCN (bSpNCN)」を提案する.

双方向予測誤差の導入

SpNCN のフレームワークに bPC の双方向予測の概念を導入する. これにより, 各層のニューロンはトップダウン (生成的) とボトムアップ (識別的) の両方の予測誤差を計算する.

■生成的予測誤差 (Top-down):

$$\mathbf{z}_{gen}^l = \mathbf{W}^{l+1} \cdot \mathbf{s}^{l+1}(t) \quad (22)$$

$$\mathbf{e}_{gen}^l(t) = \alpha_{gen}(\mathbf{z}^l(t) - \mathbf{z}_{gen}^l) \quad (23)$$

■識別的予測誤差 (Bottom-up):

$$\mathbf{z}_{disc}^l = \mathbf{V}^{l-1} \cdot \mathbf{s}^{l-1}(t) \quad (24)$$

$$\mathbf{e}_{disc}^l(t) = \alpha_{disc}(\mathbf{z}^l(t) - \mathbf{z}_{disc}^l) \quad (25)$$

統合された入力電流ダイナミクス

入力電流 $\mathbf{j}^l(t)$ の更新式に, 上記 2 種類の予測誤差と, それぞれの誤差をフィードバックするシナプスの重み行列 ($\mathbf{E}_{gen}, \mathbf{E}_{disc}$) からの入力を統合する.

中間層 l :

$$\tau_j \frac{\partial \mathbf{j}^l(t)}{\partial t} = -\kappa_j \mathbf{j}^l(t) + \phi_e \left(-\mathbf{e}_{gen}^l(t) - \mathbf{e}_{disc}^l(t) + \mathbf{E}_{gen}^l \cdot \mathbf{e}_{gen}^{l-1}(t) + \mathbf{E}_{disc}^l \cdot \mathbf{e}_{disc}^{l+1}(t) \right) \quad (26)$$

第 1 層:

$$\tau_j \frac{\partial \mathbf{j}^1(t)}{\partial t} = -\kappa_j \mathbf{j}^1(t) + \phi_e \left(-\mathbf{e}_{gen}^1(t) + \mathbf{E}_{gen}^1 \cdot \mathbf{e}_{gen}^0(t) + \mathbf{E}_{disc}^1 \cdot \mathbf{e}_{disc}^2(t) \right) \quad (27)$$

最終層 L :

$$\tau_j \frac{\partial \mathbf{J}^L(t)}{\partial t} = -\kappa_J \mathbf{J}^L(t) + \phi_e \left(-\mathbf{e}_{disc}^L(t) + \mathbf{E}_{gen}^L \cdot \mathbf{e}^{L-1}(t) \right) \quad (28)$$

ただし, $\mathbf{e}_{gen}^0(t)$ は, 入力層 (データ) に対するトップダウン予測誤差である. これらより, Δt に対する $\mathbf{j}^l(t)$ の更新則は次のようになる.

中間層 l :

$$\mathbf{j}^l(t + \Delta t) = \mathbf{j}^l(t) + \frac{\Delta t}{\tau_j} \left(-\kappa_j \mathbf{j}^l(t) + \phi_e \left(-\mathbf{e}_{gen}^l(t) - \mathbf{e}_{disc}^l(t) + \mathbf{E}_{gen}^l \cdot \mathbf{e}_{gen}^{l-1}(t) + \mathbf{E}_{disc}^l \cdot \mathbf{e}_{disc}^{l+1}(t) \right) \right) \quad (29)$$

第 1 層 :

$$\mathbf{j}^1(t + \Delta t) = \mathbf{j}^1(t) + \frac{\Delta t}{\tau_j} \left(-\kappa_j \mathbf{j}^1(t) + \phi_e \left(-\mathbf{e}_{gen}^1(t) + \mathbf{E}_{gen}^1 \cdot \mathbf{e}_{gen}^0(t) + \mathbf{E}_{disc}^1 \cdot \mathbf{e}_{disc}^2(t) \right) \right) \quad (30)$$

最終層 L :

$$\mathbf{j}^L(t + \Delta t) = \mathbf{j}^L(t) + \frac{\Delta t}{\tau_j} \left(-\kappa_j \mathbf{j}^L(t) + \phi_e \left(-\mathbf{e}_{disc}^L(t) + \mathbf{E}_{gen}^L \cdot \mathbf{e}_{gen}^{L-1}(t) \right) \right) \quad (31)$$

統合された学習則

生成過程 (トップダウン) と識別過程 (ボトムアップ) の各重みは, それぞれの予測誤差を用いて勾配降下法で局所的に更新される.

■生成過程の重み更新:

$$\Delta \mathbf{W}^l = \mathbf{e}_{gen}^{l-1}(t) \cdot (\mathbf{s}^l(t))^\top \quad (32)$$

$$\Delta \mathbf{E}_{gen}^l = \beta \left(\mathbf{s}^l(t) \cdot (\mathbf{e}_{gen}^{l-1}(t))^\top \right) \quad (33)$$

■識別過程の重み更新:

$$\Delta \mathbf{V}^l = \mathbf{e}_{disc}^{l+1}(t) \cdot (\mathbf{s}^l(t))^\top \quad (34)$$

$$\Delta \mathbf{E}_{disc}^l = \beta \left(\mathbf{s}^l(t) \cdot (\mathbf{e}_{disc}^{l+1}(t))^\top \right) \quad (35)$$

この統合により, bPC の強力な表現学習能力を, SNN のエネルギー効率の高い計算基盤上で実現することを目指す.

5 現在の状況と今後について

5.1 現在の状況

現在, 提案手法である bSpNCN モデルの実装に取り組んでいる. 関連研究論文ではモデルの理論的な枠組みは示されているが, 実装コードは公開されていないため, 数式やアルゴリズムを基に自力で実装を進めている段階である. 双方向の誤差計算やスパイクベースの非同期な更新など, 複雑な要素が多く, 実装には苦戦しているが, モデルの動作原理の理解を深めながら進めている. 実装には pytorch で SNN を実装できる Bindsnet というライブラリを用いている. Bindsnet はコンポーネントとして Nodes (ニューロン), Connection (シナプス結合), Learning rule (学習則) そしてそれらを統括する Network から成る. コンポーネントを提案した bSpNCN に合わせて定義し, 実験に合わせて Network を構築する必要がある.

5.2 今後の計画

実装が完了次第、モデルの有効性を検証するための実験を論文 [1], [2] 同様行う。

1. **データセット:** まずは標準的な画像データセットである MNIST を用いる。
2. **教師あり学習:** MNIST データセットを用いた画像分類タスクを実施し、モデルの識別性能を評価する。ベースラインとして、元の SpNCN や bPC モデルの性能と比較・検討する。
3. **教師なし学習:** 教師ラベルを用いずにモデルを学習させ、入力画像を再構成・生成するタスクを実施し、モデルの表現学習能力を評価する。
4. **性能評価:** 分類精度や画像生成の品質に加え、計算中のスパイク発火頻度を計測することで、モデルのエネルギー効率についても評価を行う。

参考文献

- [1] Gaspard Oliviers, Mufeng Tang, Rafal Bogacz. (2025): Bidirectional predictive coding, <https://arxiv.org/abs/2505.23415>.
- [2] Alexander Ororbia. (2023): Spiking neural predictive coding for continually learning from data streams, Neurocomputing Vol.544.
- [3] Hiroshi Araki. (2021): ゼロから学ぶスパイクニューラルネットワーク, <https://snn.hirrlab.net/>