

AI エッジコンピューティングにおけるニューロモルフィック強化学習の研究

研究代表者 本 木 実 熊本高専 電子情報通信工学科 教授

1 研究背景と目的

Society5.0 の実現と 2030 年に向けた SDGs 実現に近づくため、AI 技術はますます必要性を増し、情報通信インフラにおいて、クラウド AI の整備により、医療、福祉など様々な AI サービスが実現している。だがクラウド AI のみでは、IoT として組み込み電子機器を端末として利用する際、通信量、遅延、省電力性に問題がある。そこで「エッジ AI」としてエッジ側（末端側）の AI チップ（AI デバイス）が巨大 IT 企業、スタートアップ企業から提供されている。しかし、これらエッジ AI は推論のみのアクセラレータであり学習機能は搭載されていないものが殆どである。我々はこれまで、実環境学習性、リアルタイム性、省電力性（環境問題対応）を有する「エッジ学習 AI」として、SAM-SNN（Spike Accumulation and Modulation – Spiking Neural Network）を用いたエッジ学習 AI を実現した[1-3]。

しかしながら、現状の本エッジ学習 AI は、教師あり学習であるため、その端末で、環境適応のための「教師信号」の収集が必要であるという課題がある。そこで本研究調査では、SAM-SNN に強化学習のアルゴリズムを導入し、教師信号（ベクトル値）ではなく、環境からの評価値・報酬値（スカラー値）から学習する研究を行う。すなわちシステムの出力がどの程度良いかさえわかれば、自律的に環境に適応した学習「オンチップ自律学習」を行える「エッジ自律学習 AI」を見出す。

本研究の目的は、具体的なモデルとして、多層 SAM-SNN 及び、SAM ニューロンを利用した LSM (Liquid State Machine)（リザーバーコンピューティングの一種）の 2 種類のモデルについて、Actor-Critic 型の強化学習モデルを形成し、これらのモデルによる自律学習が可能かどうかを検証し、性能を評価することである。タスクとして「目標到達タスク」を設定し、計算機シミュレーションを用いた実験について報告する。

2 多層 SAM-SNN による Actor-Critic 強化学習モデル

2-1 SAM ニューロンモデル

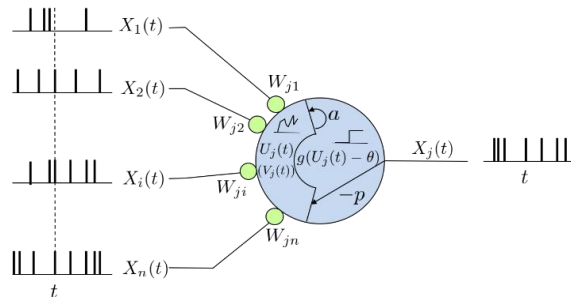


図 1 SAM ニューロンモデル

SAM ニューロンモデルの動作は以下の通りである。時刻 t において、 j 番目のニューロンが、 i 番目の入力パルス $X_i(t) \in \mathbf{B} = \{0,1\}$ を受け取ると、結合荷重 $W_{ij} \in \mathbf{R}$ との積をとり、 j 番目のニューロンの内部電位 $U_j(t)$ に加算される。この動作は、 j 番目のニューロンの入力数 n_1 だけ繰り返し、内部電位に累積されてゆく。また、内部電位 $U_j(t)$ には、前時刻の内部電位 $V_j(t-1) \in \mathbf{R}$ に減衰定数 $a \in \mathbf{R}$ を乗じた電位が残っていると考える。よって、内部電位 $U_j(t)$ は次式のように表される。

$$U_j(t) = \sum_{i=1}^{n_1} W_{ij} X_i(t) + a V_j(t-1) \quad (1)$$

ニューロンはこの内部電位 $U_j(t)$ を活性化関数 $g()$ を通し、出力 $X_j(t) \in \mathbf{B}$ を出す。活性化関数 $g()$ は、閾値 $\theta \in \mathbf{R}$

より小さければ 0, 大きければ 1 を出力するステップ関数である.

$$X_j(t) = g(U_j(t) - \theta) \quad (2)$$

$$g(u) = \begin{cases} 1 & (u \geq 0) \\ 0 & (u < 0) \end{cases} \quad (3)$$

さらに, 内部電位は発火により $p \in \mathbf{B}$ だけ減少するとし,

$$V_j(t) = U_j(t) - pX_j(t) \quad (4)$$

とする.

SAM ニューロンモデルは, (1) 時間が離散時間で表現されていること, (2) 発火時に p だけ減少すること, (3) SAM ニューロンには不応期が設定されていないこと (p で代替) の 3 点を除けば LIF ニューロンモデルと等価である. FPGA などの回路化に際して, サンプル時間 (離散時刻幅) を大きく取っても, LIF ニューロンよりも細やかな表現が可能である [7].

2-2 多層 SAM-SNN による Actor-Critic モデル

SAM ニューロンモデルを多層に結合させることで, 単層では表現できない XOR などの線形分離不可能な写像を表現できるようになる. これを多層 SAM-SNN (SAM-Spiking Neural Network) と呼ぶ.

3 層の場合で説明する. 中間層 (第 2 層) と出力層 (第 3 層) のニューロンは, 前節で説明した SAM ニューロンをそれぞれ層状に全結合する. それぞれの内部電位の初期値は 0 とする.

個々のニューロンには, 重み $\mathbf{W}$ と閾値 θ が設定されており, これらを学習パラメータとして可変させる.

学習は誤差逆伝搬法と同様であり, かつ微分不可能な演算を工夫し, かつ中間層への教師スパイクを設定することで実現している [3]. 損失 E を出力スパイクと教師スパイクの平均二乗誤差

$$E \triangleq \frac{1}{N} \sum_{P=1}^N \sum_{t=1}^{t_N} \frac{1}{2} \sum_{k=1}^{n_3} \left(X_{P,k}^{(3)}(t) - T_{P,k}^{(3)}(t) \right)^2, \quad (5)$$

とし, 勾配法により重み $\mathbf{W}$ と閾値 θ を変更する. $\mathbf{W}$ は,

$$W_{kj}^{(3)} \leftarrow W_{kj}^{(3)} - \eta \frac{\partial E}{\partial W_{kj}^{(3)}}, \quad (6)$$

$$\frac{\partial E}{\partial W_{kj}^{(3)}} = \sum_{t=1}^{t_N} \left(X_k^{(3)}(t) - T_k^{(3)}(t) \right) \frac{\partial X_k^{(3)}(t)}{\partial W_{kj}^{(3)}} \quad (7)$$

とし, 結果として, 下記のように学習する. 山括弧 $\langle \rangle$ は, 層の番号を示す.

$$\frac{\partial E(t)}{\partial W_{kj}^{(3)}(t)} = \left(X_k^{(3)}(t) - T_k^{(3)}(t) \right) g' \left(u_k^{(3)}(t) \right) H_{kj}^{(3)}(t), \quad (8)$$

$$H_{kj}^{(3)}(t) = aH_{kj}^{(3)}(t-1) + X_j^{(2)}(t). \quad (9)$$

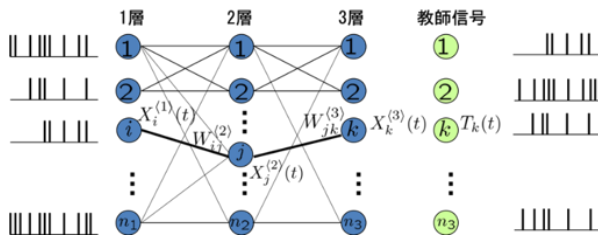


図 2 多層 SAM ニューラルネット (SAM-SNN)

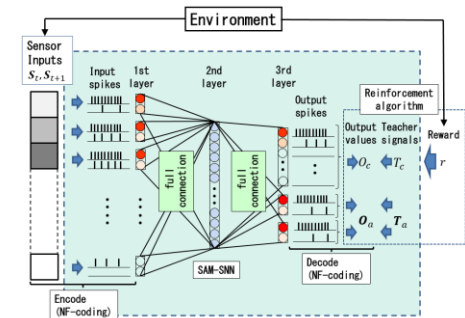


図 3 SAM-SNN による Actor-Critic モデル

多層 SAM-SNN を用いた強化学習のモデルを図 3 に示す．オープンな環境かつ入出力が連続値でも動作する オンチップ自律学習回路を目指すため，Actor-Critic 型強化学習モデルを採用する．具体的には柴田らが提案した手法を用いる[7]．

本モデルでは，まず，エージェントは現在の時刻 t での周囲の状態（センサ信号） $\mathbf{S}_t$ を観測し，ニューラル ネットの入力層に入れる．出力層では，クリティック用の出力 $O_c(\mathbf{S}_t)$ とアクター用の出力 $\mathbf{O}_a(\mathbf{S}_t)$ に分けて出力する． $\mathbf{O}_a(\mathbf{S}_t)$ に試行錯誤の乱数ベクトル $\mathbf{rnd}_t$ を足したものを動作信号として実際に動作させ，その際新たに得られた報酬 r_{t+1} ，および，センサ信号 $\mathbf{S}_{t+1}$ を入力して得られた Critic の出力 $O_c(\mathbf{S}_{t+1})$ を用いて，Critic および Actor の教師信号 T_c, T_a として学習させる．

2-3 シミュレーション

(1) 問題設定とモデル

問題設定としては目標到達タスクを選択した．まず，エリアを 2.5×2.5 に設定（y 軸に 0～2.5，x 軸に

$$T_c = O_c(\mathbf{S}_t) + \delta_t = r_{t+1} + \gamma O_c(\mathbf{S}_{t+1}) \quad (10)$$

$$T_a = \mathbf{O}_a(\mathbf{S}_t) + \alpha \delta_t \mathbf{rnd}_t \quad (11)$$

$$\delta_t = r_{t+1} + \gamma O_c(\mathbf{S}_{t+1}) - O_c(\mathbf{S}_t) \quad (12)$$

-1.25～1.25) し，その中に，視覚センサを持ったロボットと，その前方に目標を配置する．下半分にロボットの始点の位置と向きをランダムに設定する．ロボットの視覚センサは，前方 180° の視界を 24 等分し，その中に写り込む目標の割合を 0.1 から 0.9 の値でセルの中に入れる．ロボットは，この視覚センサからの情報から左右のホイールの動作を決定する．動作を重ねた結果，目標に到達したときに正の報酬 1.0 を与える．ロボットが目標に到達する，壁に当たる，あるいは最大ステップ数動いてしまうまでを 1 エピソードとする．エピソードごとに目標とロボットはランダムな位置に出現させる．

SAM-SNN の構成として，入力層のニューロン数は 48 個，中間層は n_2 個，出力層は $n_c \times 6$ 個（Critic, Actor 左, Actor 右でそれぞれ正，負の数）とした．SAM-SNN への入力としては，センサ入力値をしきい値 0.11 を用いて 0 か 1 として，すべて 0 のスパイクか，すべて 1 のスパイクとした．出力側は，NF-coding と呼ぶスパイクコーディング（発火率コーディングを基にした手法）により教師信号値をエンコードした値を教師スパイクとする．ニューロン数 $n_c = 2$ ，時間幅 $t_c = 10$ とすると，数値表現精度は $\frac{1}{n_c \times t_c} = 0.05$ である．

シミュレーションでは，学習を進めながら，途中途中でテストを行った．このときのテストのエピソード数は 200（ロボットの初期位置 5 か所 $\times$ 向き 8 方向 $\times$ 目標の位置 5 か所）とした．学習時にモデルの出力の移動量だけでロボットを動かした場合，ロボットの学習する情報がモデルに依存してしまい，モデルの出力がない状況に対しては学習が行えない．そのため学習時にはモデルの出力以外の状況を探索するための乱数（(11) 式の $\mathbf{rnd}_t$ ）を移動量に加算しており，テスト時にはこの探索用の乱数を除去して，モデルの出力のみで目標に到達するような学習ができているかを確認した．

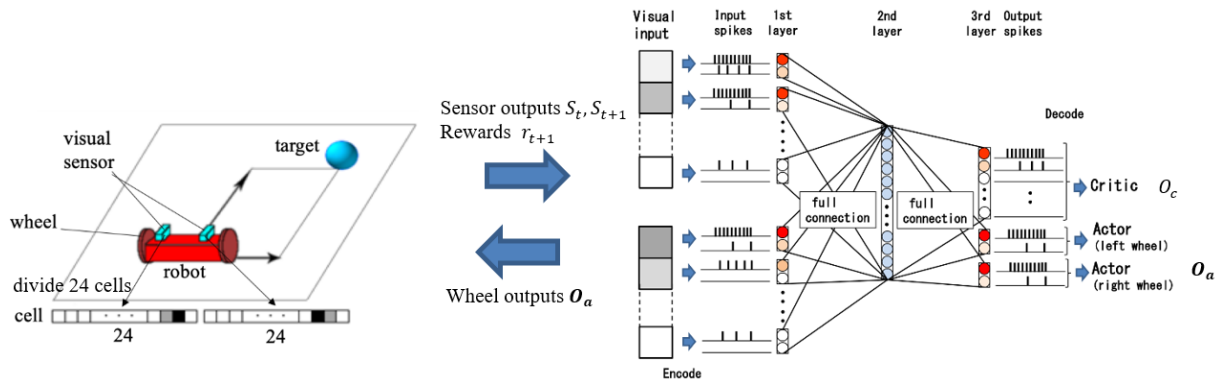


図4 シミュレーションの概要図

(2) 結果

実験は、SAM-SNN の比較対象として通常の MLP (MultiLayer-Perceptron) についても同じタスクをそれぞれ 20 回行った。各実験条件を表 1 に、学習およびテストの結果を図 5 に示す。MLP (図 5 右) ではテスト成功回数も非常に高く、また学習も安定している。

一方、SAM-SNN の方は、試行例の 1 つでは、テスト成功 195 回を達成した例があったものの、その後のテストではテスト成功回数が落ちている。また、20 回の試行において、平均成功回数も非常に低い結果となった。しかしながら、テスト 200 回すべて成功した場合も存在したため、本タスクでの SAM-SNN での学習が成功したと言える。ただし、その成功率と安定性には課題がある。

表 1 実験条件パラメータ値

条件	MLP	SAM-SNN
$n_1-n_2-n_3$	48-40-3	48-40-196
ネット構造	Linear-ReLU-Linear-Tanh	SAMSNN-SAMSNN
学習率	0.55	$2^{-5} / 2^{-10}$ (η/t)
a/p	なし	$2^{-1} / 2^{-4}$
n_c/t_c	なし	2 / 10
γ (強化学習)	0.99	0.99

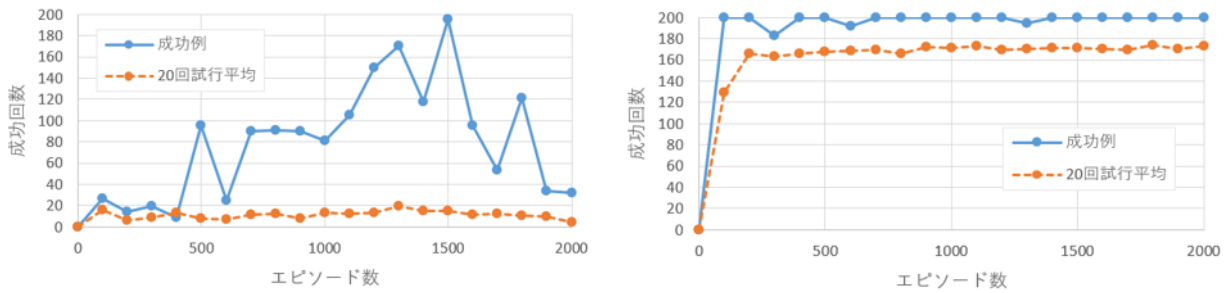


図5 テスト成功回数 (多層 SAM-SNN (左), MLP (右))

(3) 分析および考察

前節で MLP に比べて SAM-SNN でのテスト成功回数の低さと不安定性が明らかとなった。本節では、その原因の分析を行う。

まず、どのような目標到達がなされているかをロボットの軌道をプロットした (図 6)。MLP ではテスト開始後すぐに目標が見つからない状況になると、その場で回転し始めて目標をセンサ内に収めるまで回る。その後、目標をセンサでとらえたらすぐに動き始め目標に対してほぼ最短距離で動くことができている。一方、SAM-SNN での成功例 (図 6 右) では、開始後すぐに目標が見つからない場合は、少し動きながら回転しつつ、目標をセンサ内にとらえたら目標の方へ移動する。このように、MLP と SAM-SNN では成功例にしても、動きの質の獲得にも違いがある。また、SAM-SNN での別の成功例では、目標に到達したものの、直線的な動きでの到達とはならなかった。

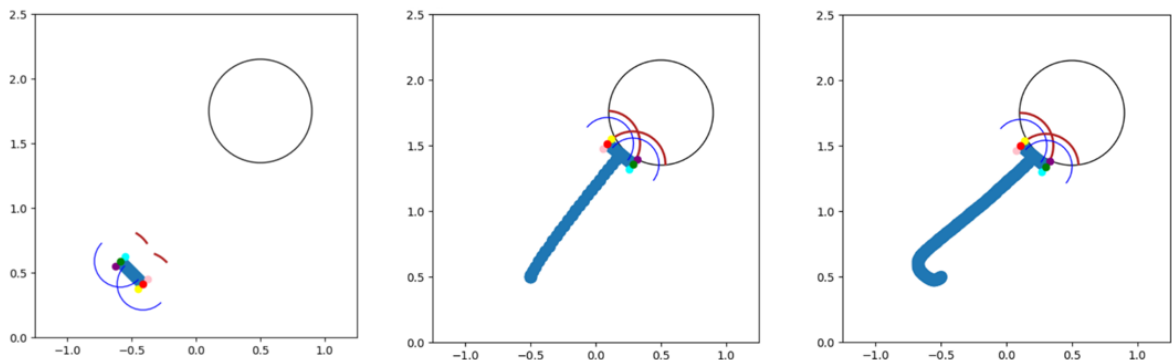


図6 ロボットの軌道 (ロボットの初期位置 (左), MLP での成功例 (中央), SAM-SNN での成功例 (右))

次に, Critic の形成過程をプロットした (図 7, 8). これは, ロボットを原点とし x, y の位置に目標を置いてセンサ入力したときの Critic の値 O_c の出力値を 0 を中心としプロットしたものである. MLP では学習の早い段階で Critic の形成が進んでおり, エピソードが進むにつれ, Critic 値が 1.0 に近い値で形成され, 目標に近い位置にある場合の Critic の値が高くなる山型のように形成されている. (割引率 $\gamma = 0.99$ と非常に高い値なので, 高い Critic 値で推移している). 一方, SAM-SNN では, エピソードが進んでも Critic の変化の幅が小さく, 値は低く, 1.0 には遠く 0.1 程度にとどまっており, Critic の形成が弱い. すなわち, Actor-Critic の強化学習が完成しているとは言い難い. この場合, 目標に到達できたのは Actor 出力 O_a のみの関数近似の実施ができたのだらうと推測される.

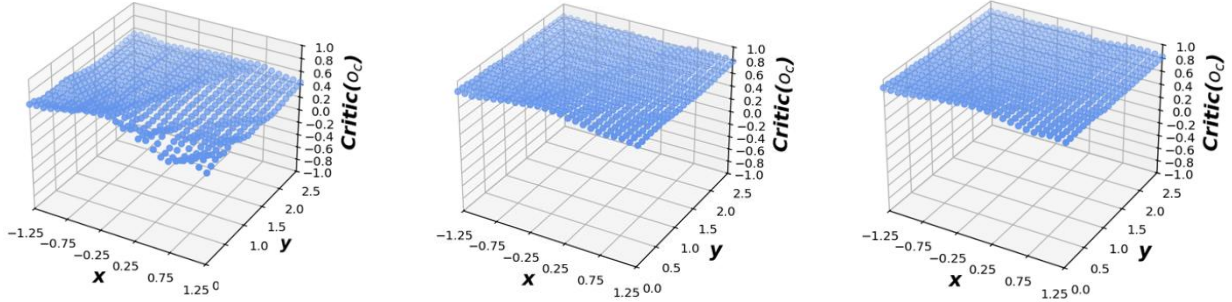


図 7 MLP での Critic の形成過程 (episode=10 (左), episode=1500 (中央), episode=2000 (右))

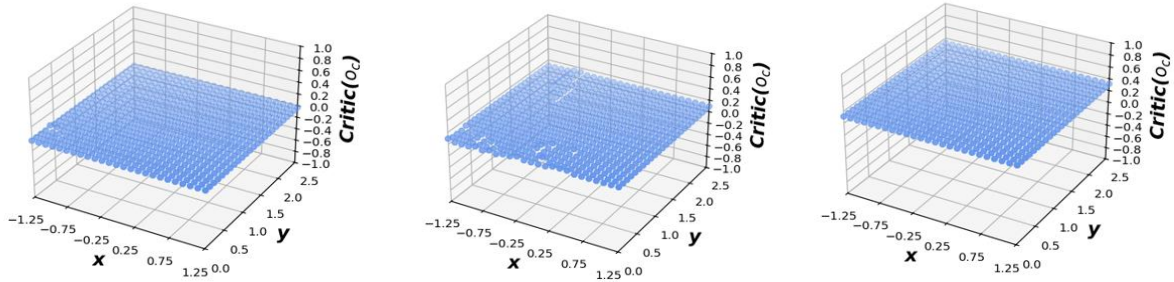


図 8 SAM-SNN での Critic の形成過程 (episode=10 (左), episode=1500 (中央), episode=2000 (右))

3 SAM-LSM による Actor-Critic 強化学習モデル

3-1 SAM-LSM による Actor-Critic モデル

LSM は, リザーバーモデルの中でもリザーバー層 (LSM ではリキッドフィルター層と呼ぶ) にスパイクニューロン, リードアウト層に形式ニューロンを用いたモデルである. 本稿では, リザーバー層に SAM スパイクニューロンモデルを用いたモデル (SAM-LSM) を提案する. SAM ニューロンは, LIF ニューロンに比べてパラメータが 1 つ多い分細やかな表現が可能のため, リザーバー層に LIF ニューロンを用いたモデルに比べて, 特に関数近似などのタスクで高い性能を示すことが期待される. 本稿ではリザーバーモデルを強化学習の価値関数近似へ利用しているため, 提案モデルが強化学習への応用として, LIF ニューロンと比べると親和性が高く, 優れていることが期待できる.

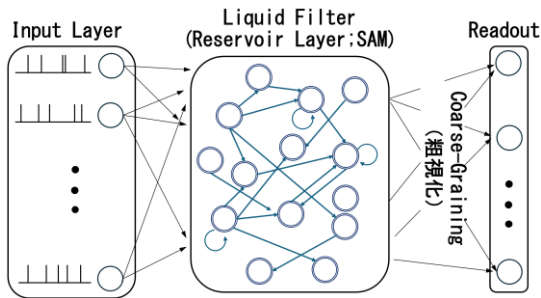


図 9 SAM ニューロンを用いた LSM モデル

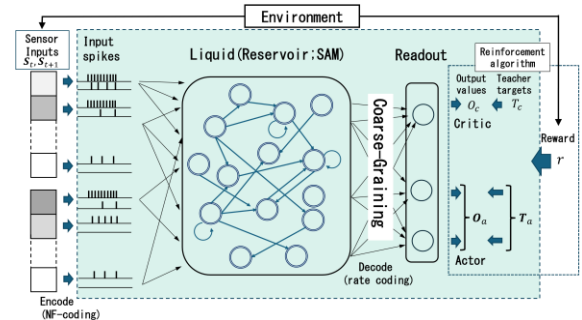


図 10 SAM-LSM による Actor-Critic モデル

ここで、リザーバー層の自己結荷重を $\mathbf{W}_h$ とすると、リザーバー層からの出力 $\mathbf{Y}(t)$ は、
で表す。 $\mathbf{W}$ と $\mathbf{W}_h$ は学習せず固定である。リザーバー層で時系列データの高次元写像を実現している。

$$\mathbf{Y}(t) = g(\mathbf{W} \cdot \mathbf{X}(t) + a\mathbf{V}(t-1) - \boldsymbol{\theta} + \mathbf{W}_h \cdot \mathbf{Y}(t-1)) \quad (13)$$

リードアウト層は、リザーバー層出力を祖視化 (Coarse-Grading) したものを入力する。祖視化はいくつかの方法も考えられるが、rate coding でデコードするものを用いる。リードアウト層の形式ニューロンの活性化関数には tanh などを用いる。重みは勾配法などで学習するため、システムに求められるタスクに適応できる。

本 SAM-LSM を Actor-Critic 型の強化学習に利用したモデルを図 10 に示す。オープンな環境かつ入出力が連続値でも動作するオンチップ自律学習を目指すため、Actor-Critic 型強化学習モデルを採用する。具体的には柴田らが提案した手法を用いる[4]。教師信号 T_c, T_a を式(10)～(12)を用いて計算し、 $O_c(S_t)$ が T_c に近づくように、 $O_a(S_t)$ が T_a に近づくように学習させる。

3-2 シミュレーション

(1) 問題設定とモデル

問題設定としては 2 章と同様の目標到達タスクを選択した。まず、エリアを 5.0×5.0 に設定 (y 軸に 0～5.0, x 軸に -2.5～2.5) する。そのエリアを y 軸方向に上方と下方の半分に分け、目標を上方のエリア内に外周壁に重ならないように配置し、視覚センサを持ったロボットを外周壁に重ならないように配置する。エピソードごとに、目標の開始位置をランダムに設定し、ロボットの始点の位置と向きをランダムに設定する。ロボットの視覚センサは、前方 180° の視界を 24 等分し、その中に写り込む目標の割合を 0.1 から 0.9 の値でセルの中に入れる。ロボットは、このセルの値から左右のホイール動作を決定する。

動作を重ねた結果、目標に到達したときに正の報酬 1.0 を与える。ロボットが目標に到達する、外周壁に当たる、あるいは最大ステップ数動いてしまうまでを 1 エピソードとする。エピソードごとに目標とロボットはランダムな位置に出現させる。

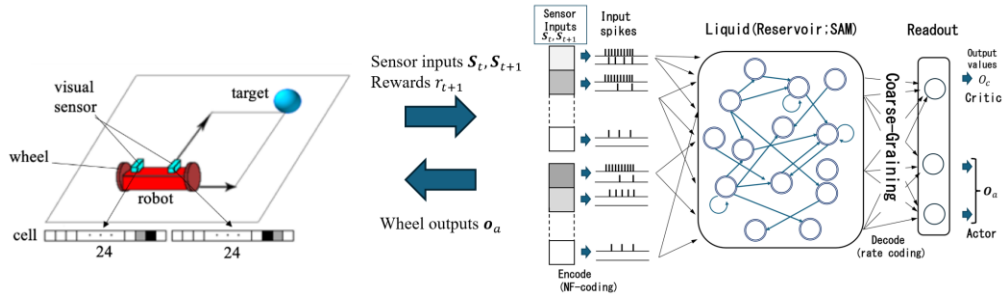


図 9 シミュレーションの概要図

SAM-LSM-RL の構成として、入力層のニューロン数は 96 個 (24×2 (左右のカメラ) $\times 2$ (n_{ci})), リザーバー層 n_2 個, 出力層は 3 個 (Critic, Actor 左, Actor 右) とした。SAM-LSM-RL への入力としては、センサ入力値を NF-coding と呼ぶスパイクコーディング (発火率コーディングを基にした手法) によりエンコードしたスパイクを入力する。NF-coding では、ニューロン数 $n_c = 2$ 、時間幅 $t_c = 10$ とすると、数値表現精度は

$\frac{1}{n_c \times t_c} = 0.05$ である。リザーバー層からリードアウト層への粗視化は発火率コーディング (rate coding) を利用した。

シミュレーションでは、学習を進めながら、途中でテストを行った、テストのエピソード数は 200 (ロボットの初期位置 5 か所 $\times$ 向き 8 方向 $\times$ 目標の位置 5 か所) として網羅的な角度で評価した。

学習時にモデルの出力の移動量だけでロボットを動かした場合、モデルの出力がない状況に対しては学習が行えない。そのため学習時には、モデルの出力以外の状況を探るための乱数 ((11) 式の $\mathbf{rnd}_t$) を移動量に加算している。テスト時には、試行錯誤の乱数ベクトルを除去して、モデルの出力のみで目標に到達するような学習ができていないかを確認した。

(2) 結果

実験は、SAM-LSM-RL の比較対象として通常の ANN モデルである MLP と多層 SAM-SNN についても同じタスク

でそれぞれ実験パラメータを変えて、無作為に 20 回試行した（それぞれ MLP-RL, SAM-SNN-RL とする）. 各実験パラメータを表 1 に、各学習エピソードにおけるテストの成功率の平均値（200 エピソード中の成功率の 20 回の平均値）を図 5 に示す. SAM-SNN-RL ではなかなか安定しないが、SAM-SNN-RL は安定して成功し、MLP-RL と同程度まで成功する結果となった.

表 2 実験パラメータ

項目	MLP-RL	SAM-LSM-RL	SAM-SNN-RL
$n_1 \cdot n_2 \cdot n_3$	48-50-3	96-50-3	48-40-192
ネット構造	Linear-ReLU-Linear-Tanh	Linear-SAM-Linear-Tanh	SAMSNN-SAMSNN
学習率	0.5	0.05	$2^{-5} / 2^{-10}$ (η/t)
a/p	なし	$1 \cdot 2^{-4} / 2^{-3}$	$2^{-1} / 2^{-4}$
$n_{ci}/n_{co}/t_c$	なし	2 / - / 25	1 / 32 × 2 / 10
γ/α	0.95/1.0	0.99/1.0	0.99/1.0
リザバー層の重み	-	スパース率 0.1 スペクトル半径 0.2	-

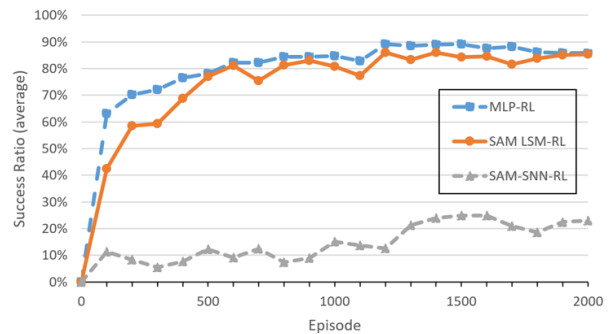


図 11 学習中のエピソード毎のテスト成功率の平均値

3-3 考察

図 11 は、各モデルの 20 回の試行の平均値を示して鳥瞰したが、更に各試行の動きを詳しく見るため、20 回の試行の成功回数を 5000 エピソードまでグラフに表した（図 12）.

SAM-LSM-RL（図 12(b), 本モデル）では、20 回のうち 1 回の試行が、なかなか成功回数が上がらず、学習に苦慮しているものの、4000 エピソードまで進むと、成功回数も約 8 割以上と学習を進められている. また、全体として、図 12(c) の MLP-RL と比較すると、成功率を高めるためにやや学習エピソード数が必要である傾向があるものの、最終的には、MLP-RL にそん色ない成功回数を示せる結果となった.

図 12(a) に示す SAM-SNN-RL においては、試行ごとのばらつきが大きく、また、一度 200 回すべて成功している結果となっているにもかかわらず、その後、また成功回数が減少してしまっている様子も見て取れる.

全体として、MLP には精度としてやや及ばないものの、SAM-LSM-RL は、MLP と同程度のニューロン数で、そん色ない精度の結果を示すことができた. 本稿では、ハードウェアとしての性能比較をするまでは行っていないが、今後、本モデルは、低コスト、低消費電力のオンチップ自律学習ができるハードウェアの一候補としての可能性を有するといえる.

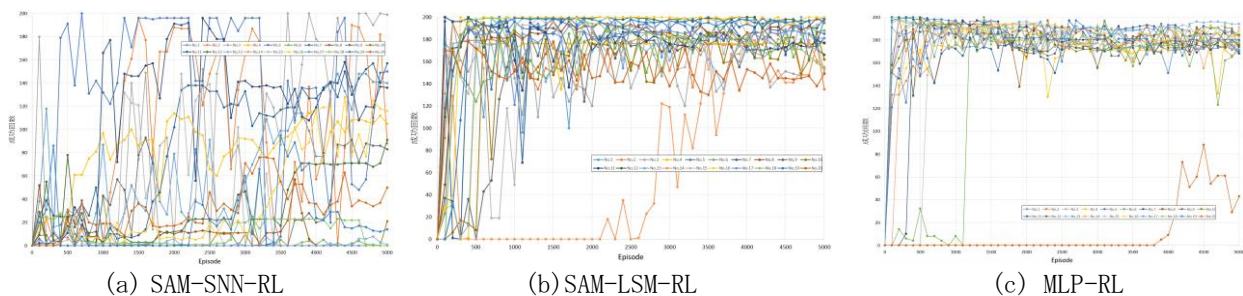


図 12 20 試行のエピソード毎の成功数

4 まとめ

本研究では、まず、オンチップ学習が可能な SAM-SNN を用い、Actor-Critic 型の強化学習を目標到達タスクにて実験した結果を報告した. その結果、比較対象の MLP と本 SAM-SNN の両者において目標到達率 100% を達成した. しかしながら、SAM-SNN では安定性と学習成功の割合において非常に劣る分析結果となり、SAM-SNN における基本的な課題が明らかとなった.

つぎに、Actor-Critic 型の強化学習の動的関数近似器として SAM ニューロンを用いた LSM を採用したモデルについて、目標到達タスクにて実験を行った. その結果、比較対象の MLP と多層 SAM-SNN と本 SAM-LSM のすべてのモデルにおいて目標到達率 100% を達成した. さらに、多層 SAM-SNN では安定性とタスク成功の割

合において非常に劣る結果であったが、本 SAM-LSM では、MLP と同等のタスク成功率を実現した。今後、より詳細な調査と分析を進め、低コスト、低消費電力の自律学習デバイスの開発を進める。

【参考文献】

- [1] 本木 実,松尾 和典,新谷 洋人,SAMスパイキングニューラルネットの FPGA 実装に関する検討, 電子情報通信学会 ニューロコンピューティング研究会, NC2017-65, pp.89-94, 2018.
- [2] Motoki, M., Shintani, H., Matsuo, K., McGinnity, TM., Function Approximation Using Multilayer SAM Spiking Neural Network, in Proceedings of IEEE 8th International Innovative Computing Technology (INTECH2018), pp.65-70, 2018.
- [3] M.Motoki et.al., An FPGA Implementation of On-Chip Trainable Multilayer SAM Spiking Neural Network, The 9th IIAE International Conference on Industrial Application Engineering 2020, pp.144-148, 2021.
- [4] 柴田克成, 強化学習とニューラルネットによる知能創発, 計測と制御, 48 巻 (2009) 1 号, pp.106-111, 2009.
- [5] 重松征史, “神経回路素子と学習・記憶”, 松本元責任編集, 日本物理学会編, 脳・心・コンピュータ, 丸善 k.k., pp.173-193, 1996.
- [6] 松本元,重松征史,市川道教,“脳研究から情報処理へ／一脳型コンピュータ開発にむけて一”,信学技法,CPSY96-108,1997.
- [7] 本木 実, SAM スパイキングニューロンモデルの頻度コーディングでの基本特性, 電子情報通信学会ニューロコンピューティング研究会 NC2021-63, pp.88-93, 2022.
- [8] Minoru Motoki, Yu Oshiro, Ryuji Waseda, and Terumitsu Nishimuta, Actor-Critic Reinforcement Learning Using On-Chip Trainable Multilayer SAM Spiking Neural Network, Proceeding of 4th International symposium of neuromorphic AI hardware, p.47, P2-15, 2022.

〈発 表 資 料〉

題 名	掲載誌・学会名等	発表年月
多層 SAM スパイキングニューラルネットによる Actor-Critic モデル	電子情報通信学会 ニューロコンピューティング研究会 NC2024-87	2025 年 3 月
Static Learning Using LSM with SAM Spiking Neural Network	Proceeding of 8th International symposium of neuromorphic AI hardware, p.26, P1-05	2026 年 3 月
SAM スパイキングニューラルネットを用いた LSM による Actor-Critic 強化学習	電子情報通信学会 ニューロコンピューティング研究会 NC2025-109	2026 年 3 月