

2023年度 春学期

## 卒 業 論 文

# SNNによるスパイクソーティング安定性の 検証および最適条件の探索

指導教員: 北野 勝則

立命館大学 情報理工学部

卒業研究3 (FO)

コース: 知能情報コース

学生証番号: 2600190482-4

氏名: JIANG Weihang

## 研究概要

スパイクソーティングは、神経科学における重要な研究テーマの一つである。スパイクソーティングは、微小電極を使用して脳のニューロンによって生成されたスパイクと呼ぶの活動電位を検出し、スパイクがどのニューロンに属することを分類するために使用される方法である。スパイクソーティングアルゴリズムは、長い間伝統的な機械学習アルゴリズムに依存してきたが、高密度多電極プローブの普及に伴い、機械学習のアルゴリズムが挑戦されている。そのような状況において、ディープニューラルネットワークの普及に伴い、ニューラルネットワークはスパイクソーティング問題において新たな解決方法を提供している。

本論文では、神経構造に基づくスパイクングニューラルネットワーク (SNN) を用いてスパイクソーティング問題を解く。このニューラルネットワークは、高度に単純化されたシミュレーションニューロンを用いて情報を学習し、実際のニューロン間の情報伝達方法であるスパイクタイミング依存可塑性 (STDP) の観点からニューロンの重みを決定し、低ノイズ環境において F-score が 0.9 以上の性能を実現できる。また、ニューロン数を改善することで、強いノイズ環境においても 0.8 の F-score を達成したことができる。その後の安定性検証においても、神経構造に基づく SNN は、スパイクタイミングやスパイク頻度が異なる環境で、強いロバスト性を示した。さらに、今回の研究によりニューラルネットワークの性能はニューロン数によって変化し、最適なニューロン数の範囲が存在することがわかった。本研究では、最適なニューロン数の範囲の一般的なルールを導出できなかったが、最適値を見つけるために、興奮性ニューロンの割合に基づく評価指標を提案した。また、ニューロン数の最適範囲を探るという課題は、今後の研究に新たな方向になる可能性がある。

キーワード：スパイクソーティング、スパイクングニューラルネットワーク (SNN)、漏れ積分発火モデル (LIF)、スパイクタイミング依存性可塑性 (STDP)

# 目次

1	研究背景	6
1.1	スパイクソーティングの研究概要	6
1.2	一般的な処理プロセス	6
1.3	ディープラーニングによるスパイクソーティングの解決策	6
1.4	SNN による新たな解決策	6
2	方法	8
2.1	シミュレーションニューロンの発火モデル：漏れ積分発火モデル（LIF）	8
2.2	重みの更新ルール：スパイクタイミング依存性可塑性（STDP）	11
2.3	スパイクニューラルネットワーク（SNN）	13
2.3.1	入力層	13
2.3.2	アテンションニューロン	14
2.3.3	中間層	14
2.3.4	出力層	14
2.4	シミュレーションデータの作り	16
3	結果	18
3.1	初期条件での性能検証および評価指標の決定	18
3.2	異なるノイズレベルでの検証	20
3.3	中間層ニューロン数における性能の影響	22
3.4	ネットワークロバスト性の検証	25
3.4.1	異なる発火タイミング	25
3.4.2	異なる発火頻度	29
4	考察	31
4.1	ニューロン数が Recall 率へ影響を与える原因	31
4.2	ニューロン数が性能へ影響を与える原因	31
4.3	SNN の将来性の考え	35
5	謝辞	36

# 目 次

1	LIF モデルの電子回路	8
2	LIF モデルの例	10
3	STDP の例	12
4	スパイキングニューラルネットワークの構造	13
5	シミュレーションデータスパイク信号例: ノイズレベル 1	16
6	シミュレーションスパイク例 1: ノイズレベル 1	17
7	中間層重み: ノイズレベル 1	18
8	中間層重み更新例: ニューロン 50 号	19
9	出力層重み: ノイズレベル 1	19
10	異なるノイズレベルでの F-score	20
11	異なるノイズレベルでの Recall 率	21
12	異なるノイズレベルでの Precision 率	21
13	中間層ニューロン数を調整した F-score	22
14	中間層ニューロン数を調整した時間	22
15	中間層ニューロン数を調整した Precision 率: ノイズレベル 1.75	23
16	中間層ニューロン数を調整した Recall 率: ノイズレベル 1.75	23
17	異なる中間層ニューロン数の F-score: ノイズレベル 1.75	24
18	異なる中間層ニューロン数の F-score の多項式フィッティング: ノイズレベル 1.75	25
19	発火タイミングが違う元スパイク信号	26
20	発火タイミングが違う中間層重み	26
21	発火タイミングが違う出力層重み	27
22	発火タイミングが違うニューロン例:1	27
23	発火タイミングが違うニューロン例:2	28
24	発火タイミングが違うニューロン例:3	28
25	発火頻度が違う元スパイク信号	29
26	発火頻度が違う中間層重み	30
27	中間層ニューロン数 25 での重み: ノイズレベル 1.75	32
28	中間層ニューロン数 26 での重み: ノイズレベル 1.75	33
29	中間層ニューロン数 30 での重み: ノイズレベル 1.75	33
30	中間層ニューロン数 34 での重み: ノイズレベル 1.75	34
31	中間層ニューロン数 35 での重み: ノイズレベル 1.75	34

# 表 目 次

1	LIF モデルシミュレーション例のパラメータ . . . . .	10
2	STDP 例のパラメータ . . . . .	11
3	入力層ニューロンのパラメータ . . . . .	13
4	アテンションニューロのパラメータ . . . . .	14
5	中間層ニューロンのパラメータ . . . . .	15
6	出力層ニューロンのパラメータ . . . . .	15
7	シミュレーションデータのパラメータ . . . . .	16
8	中間層ニューロンにおける F-score: ノイズレベル 1.75 . . . . .	24
9	興奮ニューロンの割合 . . . . .	32

# 1 研究背景

## 1.1 スパイクソーティングの研究概要

スパイクソーティングは、電気生理学的データの解析に用いられる技術である。スパイクソーティングのアルゴリズムは、脳内の1つまたは複数の電極で収集された波形の形状を使用して、1つまたは複数のニューロンの活動をバックグラウンドの電気ノイズから区別する。従来のスパイクソーティングアルゴリズムでは、主成分分析 (PCA) などの機械学習が用いられている。

これらのアルゴリズムはスパイク選別において強力な性能を示しているが、従来の機械学習アルゴリズムは、大規模で高密度の多電極デバイスを使用している今の時代では、多くの挑戦に直面している。機械学習アルゴリズムと対照的に、深層学習アルゴリズムは近年注目されている。畳み込みニューラルネットワークや、時系列データに強い感度を持つ長・短期記憶ニューラルネットワークがスパイクソーティングに新たな解決策を提供している。多くの研究者が、スパイクソーティング問題に対するディープラーニングによる解決策を研究している。

## 1.2 一般的な処理プロセス

機械学習アルゴリズムも、畳み込みニューラルネットワークのような従来のニューラルネットワークも、スパイクソーティング問題を解決する時、同じスパイクの記録とノイズ除去、スパイクの検出、特徴の抽出とクラスタリングという3つの主なタスクがある。これがスパイクソーティング問題を解く一般的なプロセスである。

具体的には、スパイクの記録とノイズ除去の段階で、ノイズは回路から、あるいはより遠くのニューロンの活動など、生体サンプル自体から発生する可能性がある。処理が行われる前に、一般的な平均化/中央値参照や信号のホワイトニングなどのノイズ低減技術が一般的に使用される。

次は波形の検出である。通常、このステップはチャンネルごとに実行され、時間間隔中にある閾値を超えた信号のデータを検出するだけである。

最後のステップは特徴抽出とクラスタリングである。スパイク検出後、整列された波形を低次元空間に投影し、クラスタリングステップを実行する必要がある。通常、この特徴抽出ステップは、ウェーブレット分解および主成分分析 (PCA) によって行われる。

## 1.3 ディープラーニングによるスパイクソーティングの解決策

機械学習アルゴリズムは通常、ステップを踏んで実行される複数の異なるアルゴリズムに分割されるが、ディープラーニングは通常、ニューラルネットワークの異なるレイヤーを使用して、これらのタスクのうちの1つ以上を達成する。このアプローチは「エンド・ツー・エンド」方式として知られており、ニューラルネットワークは生データを分類完了のスパイクに直接変換するために使用される。逆に、ディープラーニング各ニューラルネットワークが特定のステップに対応することができる。例えば、ノイズ除去の段階で、Lecoq(2021) はオートエンコーダを用いてノイズ除去問題に対処した (Deep Interpolation)。この考え方は、ある  $t$  瞬間の信号を予測するために、 $t$  瞬間の前後のサンプルを使用するディープネットワークを訓練することである。これにより、著者らは入力信号から独立したノイズを取り除くことができる。波形の検出は通常、閾値を設定することで行われる。最も重要で広く研究されている特徴抽出では、データの次元削減が主な研究対象である。オートエンコーダは低次元のポテンシャル空間を見つけるのに適しているため、波形の次元を削減するために提案されている。例えば、Wu(2019) は、検出されたスパイクから特徴を抽出するために、深度圧縮オートエンコーダを使用した。

## 1.4 SNN による新たな解決策

従来のニューラルネットワークと似ているようで異なる点として、SNN ニューラルネットワークも上記の3つの問題、すなわちスパイクの記録とノイズ除去、スパイクの検出、特徴の抽出、クラスタリングを解決する必要

があるが、上記のステップを完了するために高度に単純化されたニューロンモデルに依存し、データ転送を発火に依存している点が挙げられる。これは、情報を受け取り、受け渡すというプロセスにおいて、従来のニューラルネットワークとは根本的に異なる。

シミュレーションニューロを利用して学習し、STDP に基づく重みを決定する SNN は、多く研究者たちが今注目している。ニューロン構造と STDP 法則が脳の構造に近いと考えられ、従来のニューラルネットワークと同様の性能を持ちながら、電力性能の面で強い省エネ性を示している。それだけでなく、SNN は記録された異なるニューロンのスパイクを認識することを学習することができる。重要なのは、出力層がソーティングされたスパイクシーケンスに直接対応するスパイクシーケンスを生成することであり、このほぼリアルタイムの対応も SNN のメリットである。省エネルギー特性とリアルタイムの出力対応から、SNN は BCI アプリケーションに最適であると多くに研究者が考えている。

本研究の目的は、SNN によるスパイクソーティングロバスト性の検証と、ニューラルネットワークの構造が性能に与える影響を探索することである。

## 2 方法

### 2.1 シミュレーションニューロンの発火モデル：漏れ積分発火モデル（LIF）

積分発火モデルは、ニューロンの電氣的活動を数学的に記述するモデルのひとつである。膜電位を通してニューロンの電氣的状態を記述する。膜電位が変化する際のイオンの特定の動きを意図的に記述するのではなく、膜電位変化の2つの主な原因である膜コンデンサーと膜抵抗に直接注目するモデルである。電気抵抗の導入は、細胞膜の内外で活動するナトリウムイオンとカリウムイオンを抽象化したものであり、LIF は生物学的特性を記述する上で HH モデル（Hodgkin-Huxley モデル）より正確性に欠けるが、計算コストが少ないため、実用的なアプリケーションではより実装しやすい。

生物学の理論から明らかなように、ニューロンはリン脂質二重層からなる細胞膜に囲まれており、内部環境の安定性を確保するために絶縁体でもある。電氣的な観点から見ると、ニューロンへの電流の注入は細胞膜を充電するため、細胞膜はコンデンサーとして近似される。充電プロセスは以下の式で表される：

$$q = \int I(t) dt \quad (1)$$

膜電流としては、アクティブな伝導性を無視し、伝導性の時間に不変な成分を括り出したリーク電流のみを採用する。この時点で、細胞膜は有限の漏れ抵抗で表すことができる。

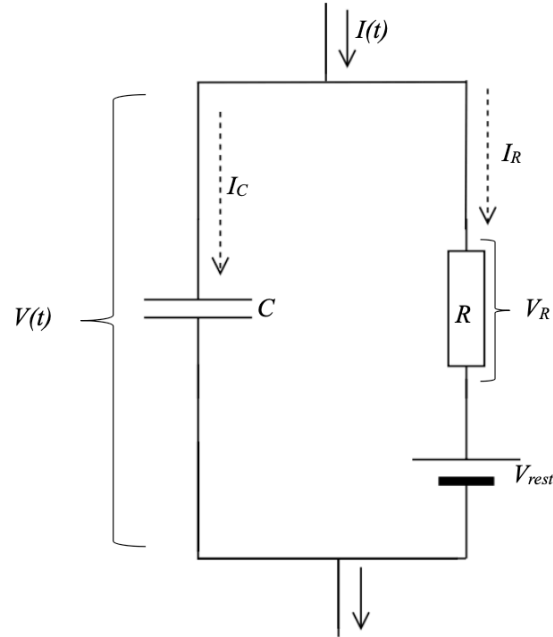


図 1: LIF モデルの電子回路

図 1 は、LIF ニューロンシミュレーションモデルの電気回路図である。回路図は、神経細胞膜の内側と外側の回路の構成を高度抽象したものである。

入力電流  $I$  による細胞膜の充電プロセスはコンデンサ  $C$  で表される。また、膜抵抗は抵抗  $R$  で表される。



キルヒホッフの法則を用いると、駆動電流  $I$  は膜コンデンサーと膜抵抗からの 2 つの成分にそれぞれ分けられ、次式に示されるようになる：

$$I = I_R + I_C \quad (2)$$

$I_R$  は、膜抵抗を通過する電流を表し、生物理論的な観点からは、膜の内外のイオンの動きによって発生する電流を表す。この成分は以下の式で計算できる：

$$I = \frac{V_R}{R} \quad (3)$$

$$V_R = V(t) - V_{rset} \quad (4)$$

$I_C$  はコンデンサーの充電であり、コンデンサーを流れる電流は以下の式で計算できる。

$$I_C = C \frac{du(t)}{d(t)} \quad (5)$$

まとめると

$$I(t) = \frac{V(t) - V_{rset}}{R} + C \frac{du(t)}{d(t)} \quad (6)$$

時定数  $\tau_m = RC$  を導入すると、次のようになる。

$$\tau_m \frac{du(t)}{d(t)} = -[V(t) - V_{rset}] + RI(t) \quad (7)$$

$\tau_m$  と  $R$  はそれぞれ膜時定数、膜抵抗と呼ばれる。

式 (7) による、ある時に膜電位が閾値電位  $V_{thr}$  に達すると、スイッチは瞬間に閉じて、コンデンサの両端の電圧はリセット電位にリセットされ、スイッチが瞬間開くものとする。したがって、そのスイッチみたいな開閉はニューロンの発火を表すことができる。

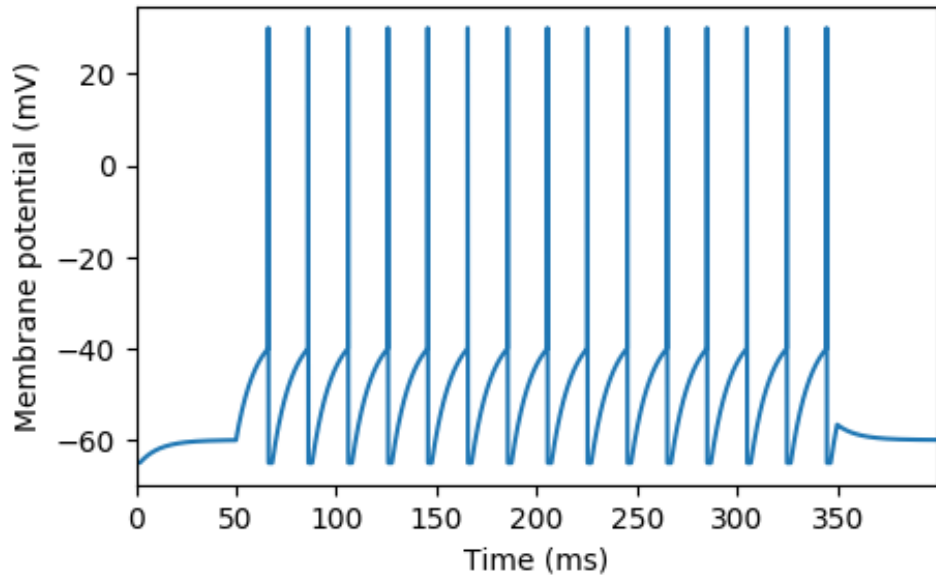


図 2: LIF モデルの例

図 2 は、LIF モデルの例である。各個 LIF のパラメータは表 1 から確認できる。

表 1: LIF モデルシミュレーション例のパラメータ

パラメータ名	パラメータ値
時間間隔 $dt$	$5e-5(s)$
シミュレーション時間 $T$	$0.4(s)$
不応期 $t_{ref}$	$2e-3(s)$
膜時定数 $\tau_m$	$1e-2(s)$
膜電位 $V$	$-60(mV)$
リセット電位 $V_{rset}$	$-65(mV)$
閾値電位 $V_{thr}$	$-40 (mV)$
ピーク電位 $V_{peak}$	$30(mv)$

図 2 と表 1 から見ると、シミュレーションニューロンは、閾値電位である  $V_{thr}$  に達すると、電位瞬間にピーク電位  $V_{peak}$  に発火し、その後即リセット電位  $V_{rset}$  にリセットする。このようなスイッチみたいな開閉作動はニューロンの発火、興奮を表すことができる。したがって、LIF モデルによって、ニューロンのスパイクが計算的にシミュレーションすることができる。

## 2.2 重みの更新ルール：スパイクタイミング依存性可塑性（STDP）

従来のニューラルネットワークの学習則は、Hebb 則から始まり、今の時代では 1986 年に提案された誤差逆伝播学習法によって使っている。しかし、SNN では、スパイク列というの離散的なデータを情報としているので、従来の学習則は使うことができない。したがって、SNN でよく使ってる学習則は、スパイク列に基づいたスパイクタイミング依存性可塑性（STDP）である。

実際の生物体内では、神経細胞は複雑な動力学系であり、静止状態、興奮状態、パルスの放出、パルスの受信、膜電位の蓄積、不応期など、さまざまな状態と行動を持っている。研究により、2つの神経細胞間のシナプス接続の強度の変化が前後神経細胞の発火時間差に関連していることがわかった。シナプス前およびシナプス後の神経細胞の発火の相対的な時間差が、神経細胞間のシナプス接続の重みの変化の方向と大きさに重要な役割を果たしている。このシナプス前後細胞の発火時間差に基づく学習則は、スパイクタイミング依存性可塑性（STDP）と呼ばれている。

具体的には、STDP ルールの原理は以下のように具体的に説明される：一定の時間内で、もしシナプス前の神経細胞が発火し、その後にシナプス後の神経細胞も発火させた場合、これらの2つのスパイクは因果関係があると見なされ、それらの間のシナプス接続は強化される。強化の程度は、発火時間差によって決まる。逆に、もしシナプス前の神経細胞がシナプス後の神経細胞よりも遅れて発火させた場合、二者間に因果関係がないと見なされ、その結果、シナプス接続は弱くなる。弱まるの程度も、発火時間差によって決まる。

数式表すと、シナプス細胞の発火時刻と後細胞の発火時刻は、それぞれ  $t_{pre}$  と  $t_{post}$  と呼ぶ。これに対して、この二つの細胞の発火時間差を  $\Delta t$  と呼ぶ。STDP 則によって、前後細胞を繋がるのシナプス重みは重み  $\Delta w$  は、 $\Delta t$  を依存し、以下の式に従って変化する。

$$\Delta w = \begin{cases} A_+ \exp(-\frac{\Delta t}{\tau_+}) & \Delta t > 0 \\ -A_- \exp(-\frac{|\Delta t|}{\tau_-}) & \Delta t < 0 \end{cases} \quad (8)$$

$A_+$  と  $A_-$  は正の定数で、学習率と同じである。

表 2: STDP 例のパラメータ

パラメータ名	パラメータ値
学習率 $A_+ = A_-$	1.05
$\tau_+ = \tau_-$	20(ms)

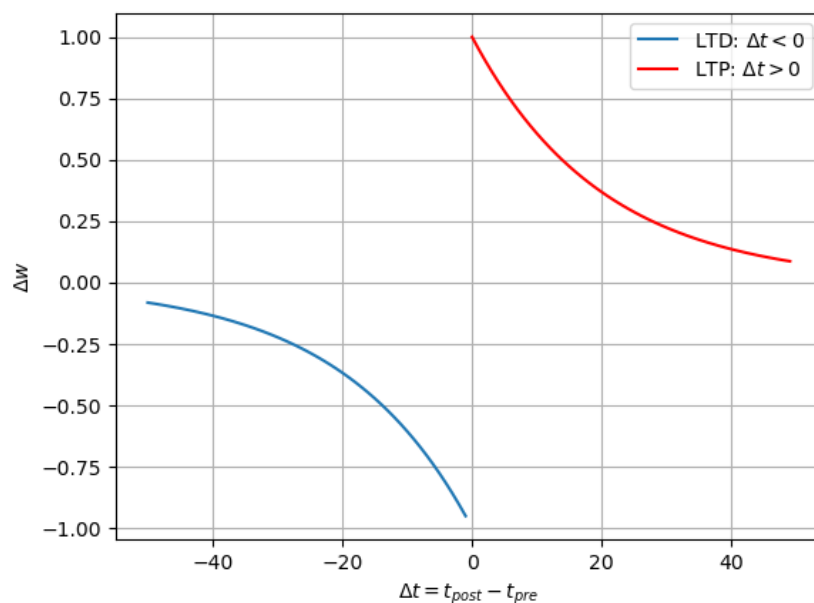


図 3: STDP の例

図 3 は、STDP ルールの例である。各個 STDP ルールのパラメータは表 2 から確認できる。

図 3 と表 2 から見ると  $\Delta t > 0$  の場合は LTP(long-term potentiation, 長期増強) を起こり、シナプス接続は強化される一方で、 $\Delta t < 0$  場合は LTD(long-term depression, 長期抑圧) を起こり、シナプス接続は弱くなる。したがって、STDP の法則によって、シナプスの重みがみまることができる。重みによって、ニューロンが学習できる。

## 2.3 スパイキングニューラルネットワーク (SNN)

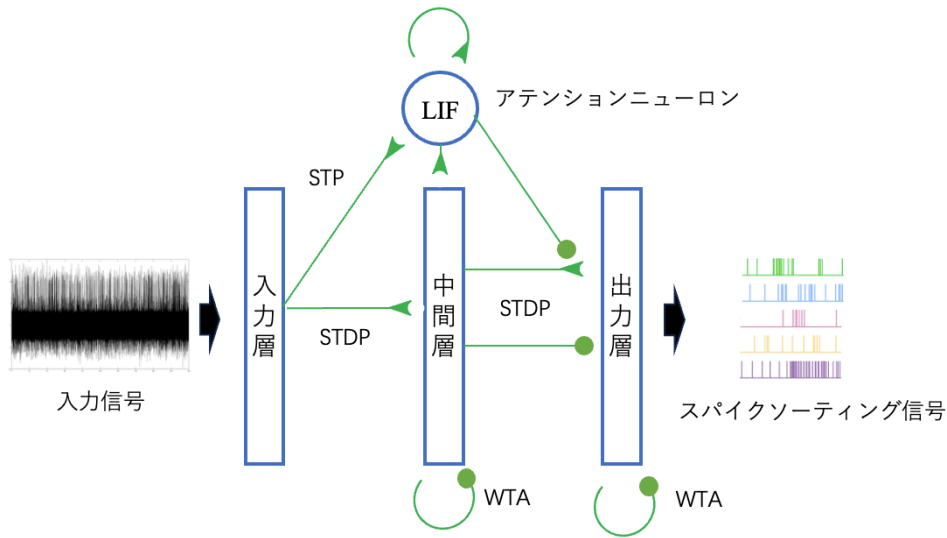


図 4: スパイキングニューラルネットワークの構造

スパイキングニューラルネットワーク (SNN) は、神経システムを模倣したニューラルネットワークの一種である。通常のニューラルネットワークとは異なり、スパイクを情報の伝達に使用する。スパイキングニューラルネットワークでは、神経細胞の状態を表現するために膜電位を使用する。神経細胞は一定の膜電位のしきい値を超えるとスパイクを生成し、シナプスを介して情報を伝達する。

本研究で使用した SNN ニューラルネットワークのアイデアは、先行研究である Bernert ら (2019) から提案された 3 層ニューラルネットワークから派生したものである。

図 4 のように、この SNN ニューラルネットワークには、入力層、中間層、出力層の 3 つの層が含まれている。ニューロンの発火をシミュレーションするために、LIF モデルを使用し、シナプスの重みを決定するために STDP 法則を使用する。これに加えて、このモデルにはスパイクのみを集中するため、一つのアテンションニューロンが含まれている。

### 2.3.1 入力層

入力層の役割は入力信号をエンコードすることである。スパイクのエンコードは閾値に依存する。入力層は 50 行 10 列の 500 個のニューロンのアレイで構成され、スライディングウィンドウ  $t_s$  の間隔で生データ全体を掃引する。各ニューロンは同じ感度区間  $\Delta V_s$  を持ち、信号区間全体をカバーするのに十分な重なりがある。元信号の値がニューロンの感度区間内に入ると、そのニューロンが発火する。この結果、入力層を通して入力信号が人工的なスパイクの列に連続的に変換され、スライディングウィンドウのたびに徐々にエンコードされる。

表 3: 入力層ニューロンのパラメータ

パラメータ名	パラメータ値
スライディングウィンドウ間隔 $t_s$	0.0125(ms)
感度区間 $\Delta V_s$	$3.5\sigma_{noise}$

### 2.3.2 アテンションニューロン

アテンションニューロン主の役目は、スパイクの検出と中間層および出力層の活性化である。

アテンションニューロンは、3層の各ニューロンを完全に連結している。入力層から入力層スパイク列を受け取り、入力層のスパイク列に基づいて元データのニューロンスパイクを検出する。

元データのニューロンスパイクを検出することができるの原理は、短期可塑性（STP）ルールである。短期可塑性を表す式は、以下である：

$$\frac{d\omega}{dt} = \frac{1}{\tau_{stp}}(1 - \omega) - \sum_s \omega f_d \delta(t - t_{pre}) \quad (9)$$

ここで  $\omega$  はシナプスの重み、 $\tau_{stp}$  は時間定数、 $f_d$  は抑圧係数、 $t_{pre}$  はシナプス前細胞のスパイク時間である。

STP ルールは、発火頻度が高いニューロンの重みを減少する機能がある。データスパイク範囲内にある入力層ニューロンは、ノイズ範囲内にある入力層ニューロンより発火頻度が低いため、その結果、ノイズの範囲内の信号をエンコードする入力層ニューロンに対応するシナプスの重みは弱まり、データスパイク範囲内の信号をエンコードする入力ニューロンに対応するシナプスの重みは強いまとなる。したがって、この法則によって、ノイズの影響が軽減され、元データのスパイクのみ学習する。

元データのスパイクを検出した後、アテンションニューロは自身のスパイクシーケンスを使って、中間層と出力層に完全に接続することで、両層のニューロンの興奮時刻を制御する。後者の2層のニューロンは、アテンションニューロン自身が発火した時のみ学習を開始させる。つまり、後者の2層のニューロンは、元データのニューロンスパイクのみを学習することを保証する。後者の2層のニューロンに学習の時刻を制御するの役割は、「アテンションニューロン」という名前の由来である。

表 4: アテンションニューロのパラメータ

パラメータ名	パラメータ値
不応期 $t_{ref}$	0
膜時定数 $\tau_m$	0.025(s)
短期可塑性定数 $\tau_{stp}$	5(ms)
抑圧係数 $f_d$	0.0161

### 2.3.3 中間層

入力層とアテンションニューロンの目標はスパイクの検出だとすると、中間層のタスクは、入力層とアテンションニューロンによって検出されたスパイク波形を学習することである。

入力層と中間層は完全連結で接続され、2層のニューロンを接続するシナプスの重みは STDP の法則によって決定される。興奮したニューロンは、WTA の法則によって他のニューロンを抑制し、全データが学習されるまで次のニューロンを学習続く。中間層のニューロン数は固定されておらず、ニューラルネットワーク全体の性能は、その後の探索でニューロン数に対して強い感度を示すが、これは結果後の考察部分で詳しく説明する。中間層のニューロンは同じで LIF モデルを使用し、中間層で生成されたスパイク列は出力層ニューロンに渡される。

### 2.3.4 出力層

出力層のタスクは、中間層で学習されたスパイクを分類することである。

出力層は 10 個の LIF ニューロンで構成され、それらは中間層と同様に完全連結で接続され、重みは STDP 則

表 5: 中間層ニューロンのパラメータ

パラメータ名	パラメータ値
不応期 $t_{ref}$	0.05(ms)
膜時定数 $\tau_m$	0.025(s)
リセット電位 $V_{reset}$	0
$\tau_+$	0.0563(ms)
$\tau_-$	0

によって決定される。初期閾値  $Th$  は 140 で、学習中閾値が低く更新する。スパイクングニューラルネットワークの利点の 1 つは、出力層が出力するスパイクシーケンスが元データから分類したスパイクシーケンスであることで、リアルタイムで分類できる。元データに 3 つのニューロンが含まれている場合、出力層は WTA 則によってちょうど 3 つのニューロンが興奮され、他のニューロンが抑制される。

表 6: 出力層ニューロンのパラメータ

パラメータ名	パラメータ値
不応期 $t_{ref}$	3(ms)
膜時定数 $\tau_m$	3(ms)
$\tau_+$	3(ms)
$\tau_-$	0.2(ms)
初期ニューロン閾値 $Th$	140
リセット電位 $V_{reset}$	$-10 * Th$

## 2.4 シミュレーションデータの作り

ネットワークの性能を検証するために、信用できるのシミュレーションデータを作る必要がある。  
シミュレーションデータの作り方は、Adamos ら (2008) から提案した方法である。

$$V(t) = A \sin\left(\frac{t - t_0}{\tau_1}\right) \exp\left(-\left(\frac{2.3548t}{\tau_2}\right)^2\right) \quad (10)$$

パラメータ  $A$  は各スパイク波形の最大値を決定し、各ニューロンの波形発生はポアソン過程に従って定義される。同時に、各ニューロンに 3ms の不応期が適用された。不応期とは、ニューロンが活動電位を発信した後、そのニューロンが再び活動電位を発信できない期間のことである。この不応期の目的は、ニューロンの生理学的特性をシミュレーションすることである。ノイズレベルの異なるシミュレーションデータを生成するために、異なる SNR(信号対雑音比) を使用した。SNR は、各個ニューロンの活動電位のピーク電位の平均  $A_{avg}$  とノイズの標準偏差  $\sigma_{noise}$  の比として定義される。

本研究では、式 (10) に従って作った Adamos(2008) から公開された 200 秒の三つのニューロンが含まれるシミュレーションデータセットを使用した。各個ニューロンのパラメータは、表 7 確認できる。また、ロバスト性検証に関する異なる発火タイミングと発火頻度のデータは、先行研究である Bernert(2019) から公開されたデータセットに基づく、発火ルールと発火頻度を修正したシミュレーションデータを使用した。

表 7: シミュレーションデータのパラメータ

ニューロン名	$A$	$\tau_1(ms)$	$\tau_2(ms)$	$t_0(ms)$
1	5	1	0.5	-0.25
2	5	1	0.5	0.25
3	10	1	0.5	-0.20

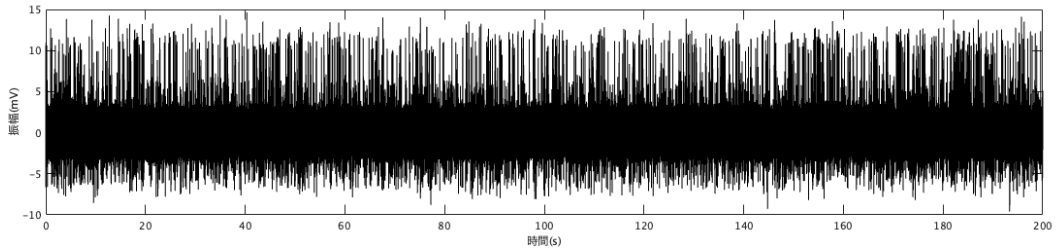


図 5: シミュレーションデータスパイク信号例: ノイズレベル 1

図 5 はシミュレーションデータスパイク信号例である。横軸が時間、縦軸は振幅である。また、ノイズレベルが 1 の時を初期条件とし、他にもノイズレベルが 0.5, 0.75, 1.50, 1.75, 2.00 の実験環境がある。



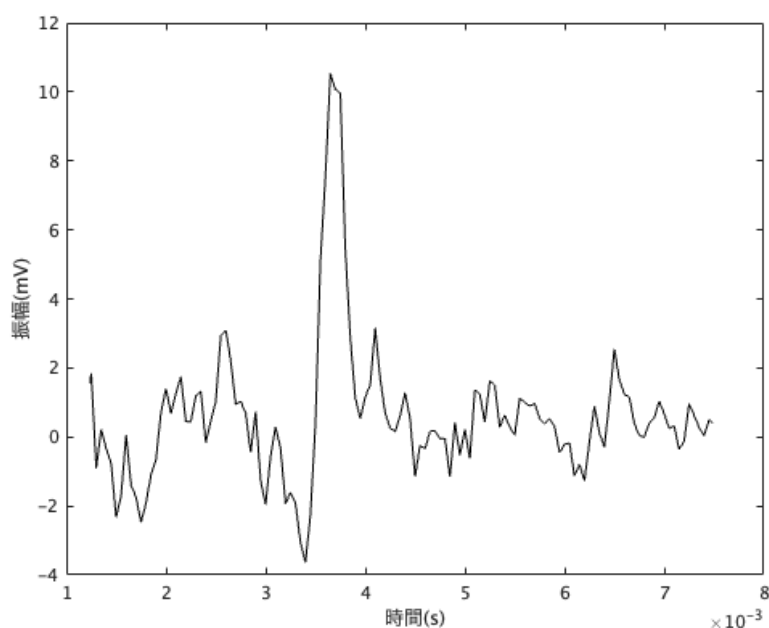


図 6: シミュレーションスパイク例 1: ノイズレベル 1

図 6 は、シミュレーションスパイクの例である。

横軸は時間、縦軸は振幅である。ニューロンは入力に応答して自らの電位を激しく変動させる。これは活動電位と呼ばれる。電気生理学では、微小電極を用いてその活動を測定する。活動電位が図 6 のように、急激に変動することがわかる。この電位の急激な変化をスパイクと呼ぶ。この電位の変化は、図 6 を通して観察することができ、シミュレーションスパイクの作りが成功したことを示している。

また、シミュレーションスパイクの波形は、表 7 のパラメーターで制御できる。パラメータ  $A$  は振幅の大きさを制御し、パラメータ  $t_0$  の正負はスパイクの傾向を制御する。

本研究では、表 7 のように、一つのニューロンがより高い振幅を持つ (3 号)、もう一つのニューロンが正負が異なる  $t_0$  値を持つ (2 号)、すなわち異なる波形傾向を持てる。これにより、ニューラルネットワークのロバスト性を総合的にテストすることができる。

### 3 結果

#### 3.1 初期条件での性能検証および評価指標の決定

最初に検証したのは、このモデルがスパイクソーティングができるのかという基本的な課題である。実験対象はシミュレーションデータである。シミュレーションデータは、ニューロンラベルもつけているので、定量的に精度指標を計算可能である。具体的には、このモデルに対しては、全てのスパイクが識別されるか、識別のスパイクが正しいか両方で評価必要がある。全てのスパイクが識別されるかは Recall 率で評価し、識別のスパイクが正しいかは Precision 率で評価する。

$$Recall = \frac{Hit}{Truth} \quad (11)$$

$$Precision = \frac{Hit}{Output} \quad (12)$$

その両方も評価することができるのは、神経科学がよく使った F-score である。

$$F-score = \frac{2 * recall * Precision}{Recall + Precision} = \frac{2 * Hit}{Recall + Precision} \quad (13)$$

実験対象として、まずは三つのニューロンがある、ノイズレベルが1の環境で初期条件とし、ネットワークの性能を検証する。

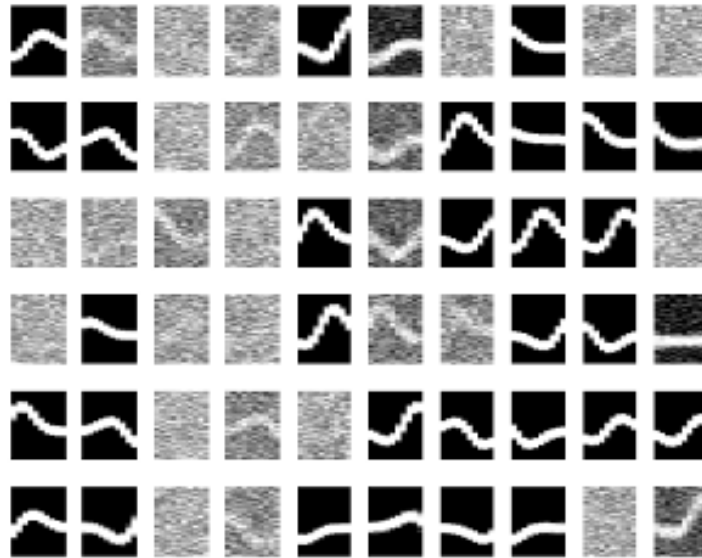


図 7: 中間層重み: ノイズレベル 1

図 7 は、中間層ニューロン重みの可視化である。各パネルが、一つの間層ニューロンに対応してる。パネルに写ってる波形は、ニューロンが敏感であるスパイク波形に対応する。画像が明らかになってるほど、その中間層ニューロンの重みが高くなってる。

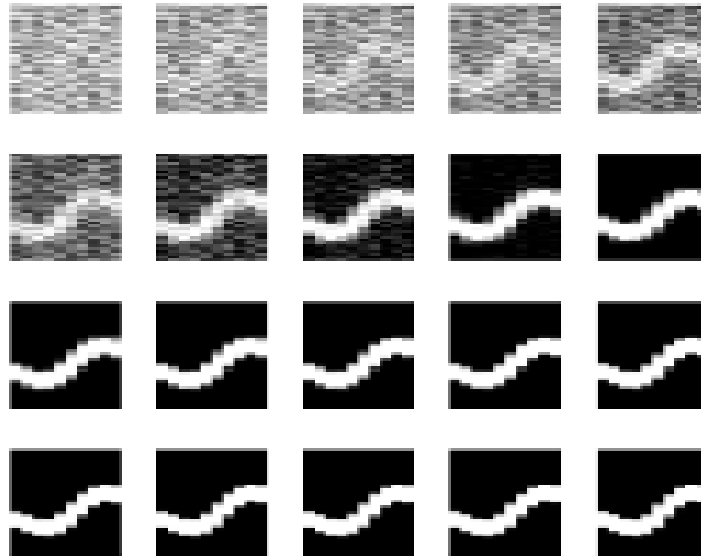


図 8: 中間層重み更新例: ニューロン 50 号

図 8 は、中間層ニューロン重みの更新プロセスである。各パネル、10 秒ずつ中間層のニューロンの重みを可視化する。図 8 から見ると、中間層ニューロンは、スパイクの波形を学習しながら、写ってる波形が明らかになる、すなわち重みが更新続ける。

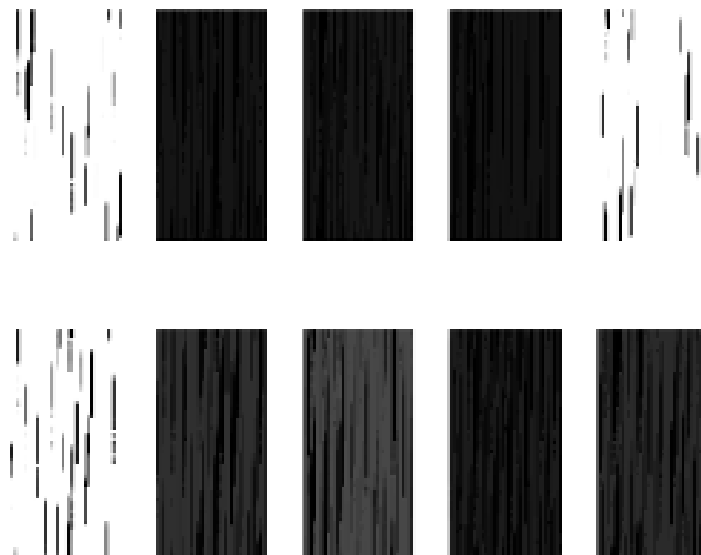


図 9: 出力層重み: ノイズレベル 1

図 9 は、出力層ニューロン重みの可視化である。各パネルが、一つの出力層ニューロンに対応してる。抑制されはニューロンは、黒に見える。図 9 から見ると、出力層は、三つのニューロンだけ抑制してない、つまりこの時は三つのニューロンが興奮し、元データと一致する。全体として、このモデルは予測通り動かす、スパイクソーティングができる。

### 3.2 異なるノイズレベルでの検証

次は異なるデータのノイズレベルでの実験である。このモデルが正しく作動することがわかったら、さまざまなノイズ環境におけるモデルの性能を調べる必要がある。使用したノイズ環境は 0.5,0.75,1.00,1.50,1.75,2.00 である。

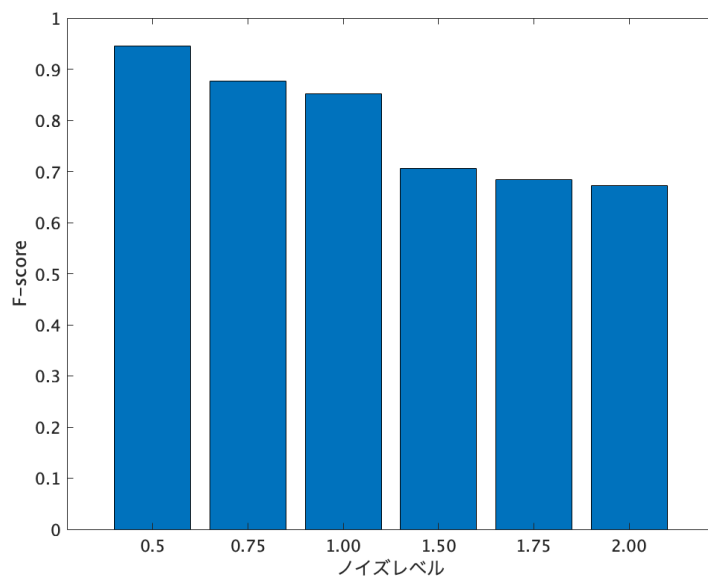


図 10: 異なるノイズレベルでの F-score

横軸はノイズレベルで、縦軸は F-score である。図 10 から見ると、全体として、ノイズレベルが低い場合はこのモデルの性能が高いが、高い場合は 0.6 ぐらいしかいない。問題点はどこにあるのかを考えて、まずは Recall 率と Precision 率両方に分けて、評価する。

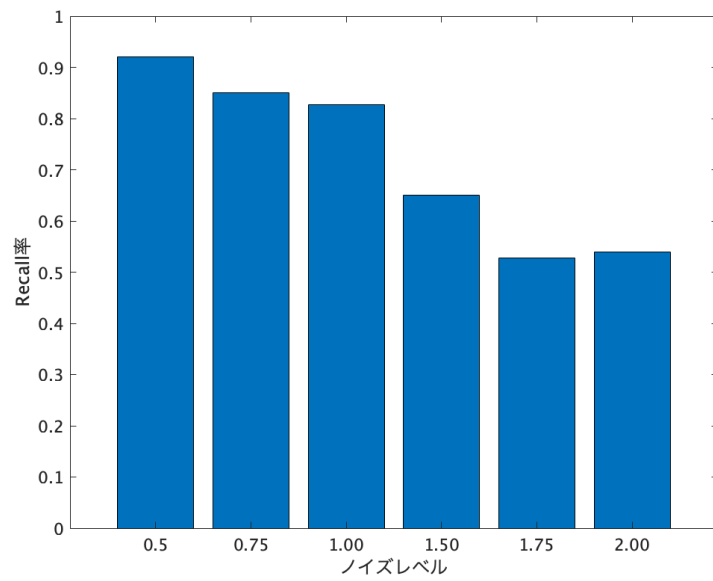


図 11: 異なるノイズレベルでの Recall 率

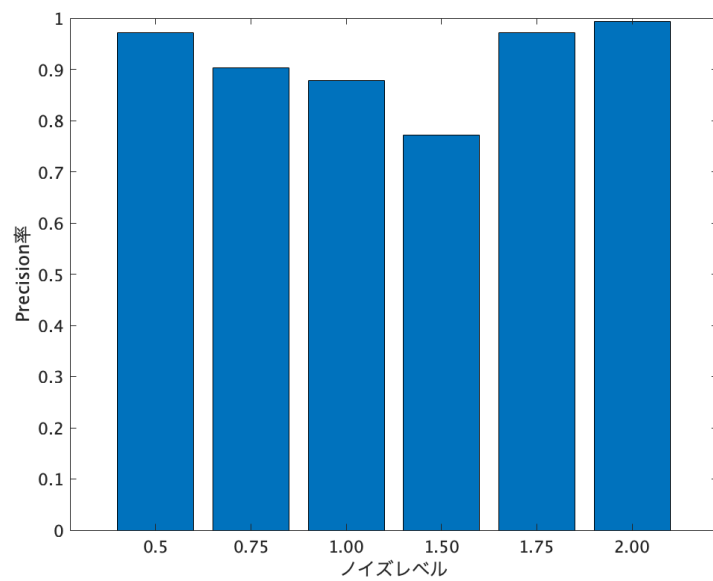


図 12: 異なるノイズレベルでの Precision 率

図 11 と図 12 から見ると、高いノイズレベルの場合でも、Precision 率が非常に高い、つまり認識したスパイクがほぼ正しく識別されたことがわかった。問題点は Recall 率である。高いノイズレベルの場合、全てのスパイクを検出することができない。

### 3.3 中間層ニューロン数における性能の影響

ニューラルネットワークは、ノイズレベルが高い状況ではすべてのスパイクを完全に検出することはできない。したがって、最初に思いつく改善策は、ニューロンの数を変えて、ニューロンの数がニューラルネットワーク全体の性能に影響するかどうかを調べることである。初期状態の中間層 60 ニューロンを 30 ニューロンと 120 ニューロンに調整し、ニューロン数の影響を探索してみる。

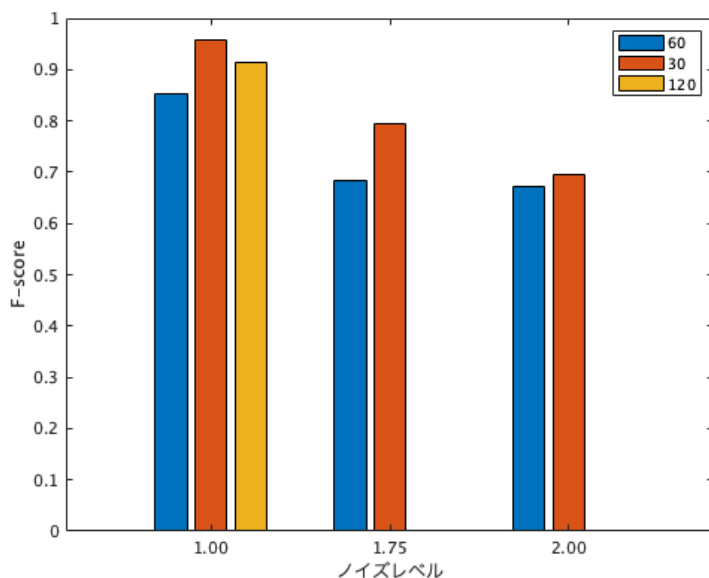


図 13: 中間層ニューロン数を調整した F-score

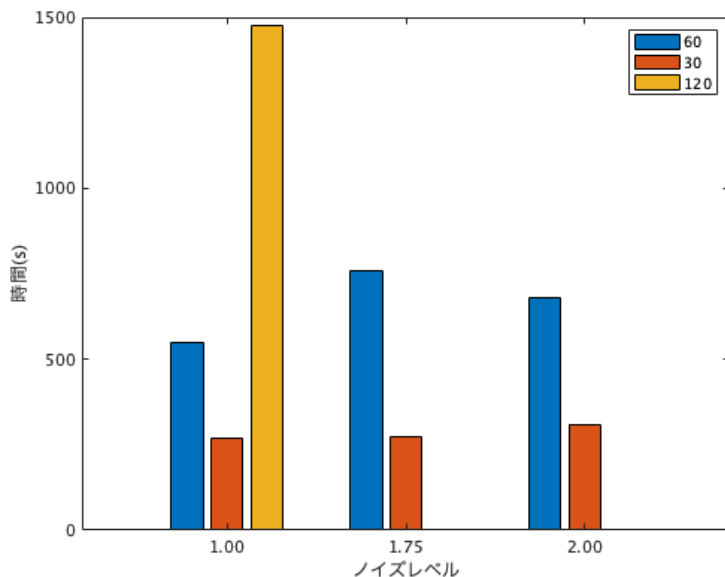


図 14: 中間層ニューロン数を調整した時間

図 13 と 14 から見ると、ニューロン数を倍にすると、性能の方は顕著に上がるではないが、非常に時間がかかる。具体的に分析すると、時間の方は単なる倍になったではない、せめても非線形で増加する。増加するの策は、

おそらく甲斐がないと思う。逆に、ニューロン数を半分にすると、性能の方は顕著に上がるの一方で、スパイクソーティングの速度も速い。したがって、ニューロン数の調整の方向は、減少するの方が甲斐があるかもしれない。

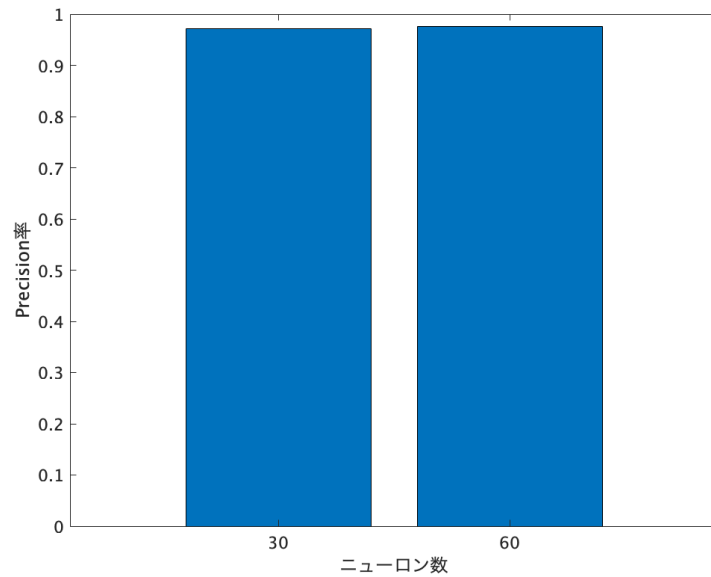


図 15: 中間層ニューロン数を調整した Precision 率: ノイズレベル 1.75

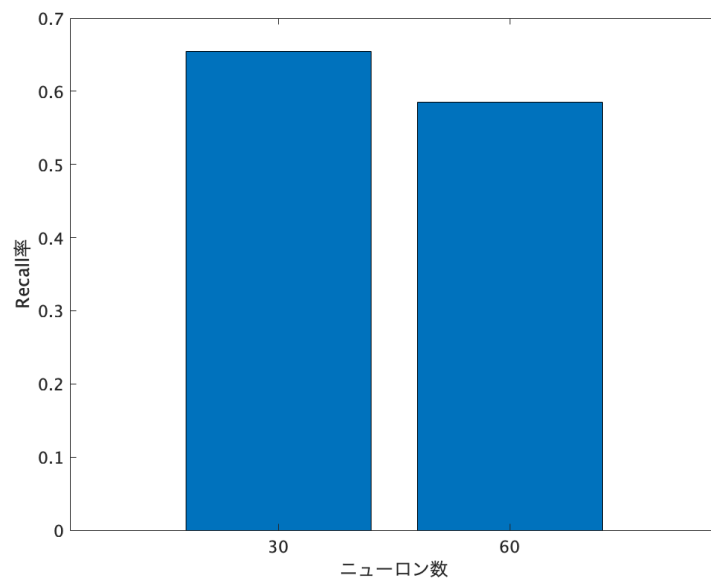


図 16: 中間層ニューロン数を調整した Recall 率: ノイズレベル 1.75

図 15 と図 16 から具体的に分析すると、ニューロン数を減少すると、Precision 率が顕著に変わらないの一方、Recall 率が、顕著に上がる。図 11 と図 12 の結論から見て、すでに問題点は Recall 率であることがわかる。したがって、ニューロン数を減少するのはこの問題点を解決できることがわかる。

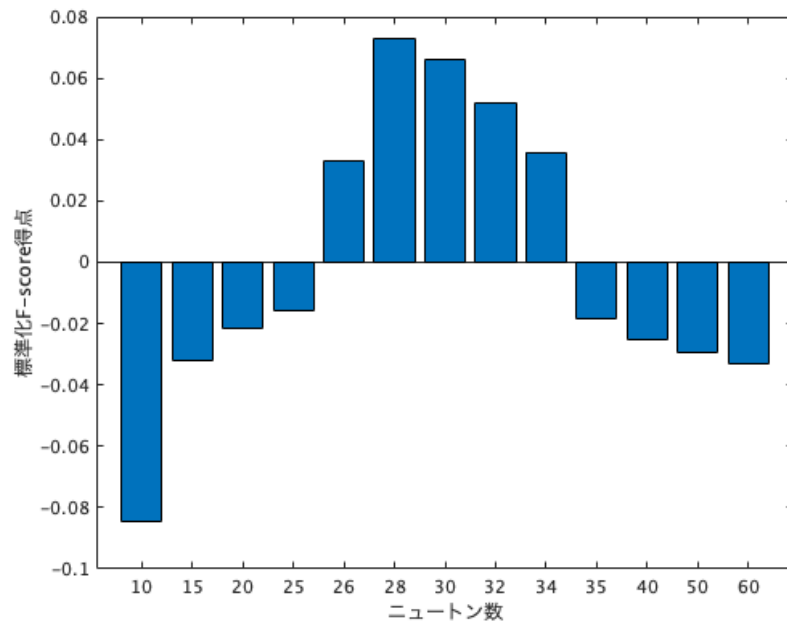


図 17: 異なる中間層ニューロン数の F-score: ノイズレベル 1.75

表 8: 中間層ニューロンにおける F-score: ノイズレベル 1.75

中間層ニューロン数 (個)	F-score
25	0.702
26	0.750
28	0.790
30	0.783
32	0.769
34	0.753
35	0.699
40	0.691
50	0.688
60	0.684

図 17 は、ニューロン数全体の考察である。答えたいのは、ニューロン数を減少して、Recall 率は改善することができるという結論の上で、さらに減少できるのか、何かの限界、閾値みたいな範囲があるのかという疑問である。30 を中心に、ニューロン数を 10,15,20,25,30,35,40,50,60 の順番でテストする。さらに、今まで最も性能が高いニューロン数 30 の近くに、26,28,30,32,34 の順番で、より詳細的にテストする。

横軸はニューロン数、縦軸は F-score を平均 0 にした標準化の結果である。表 8 の結果から具体的見ると、確かに最も高い F-score はニューロン数 30 の近くに達する。しかし、さらに劇的なのは、ニューロンの数が 34 から 35 にと 25 から 26 にわずかに変化すると、ネットワーク全体のパフォーマンスが大幅に向上する。まるで限界があるみたい、ある閾値を突破すると、性能が大きく向上することがわかる。



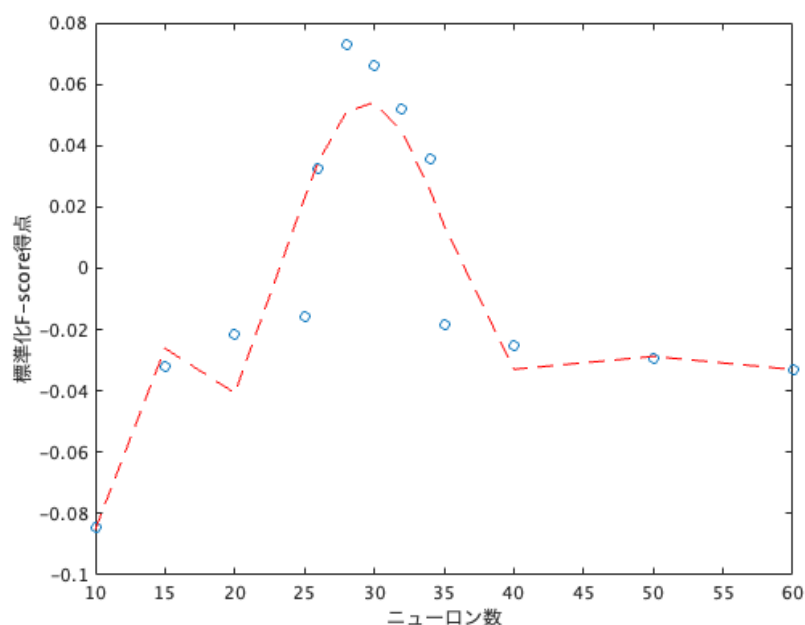


図 18: 異なる中間層ニューロン数の F-score の多項式フィッティング: ノイズレベル 1.75

図 18 は、F-score をニューロン数に対しての多項式フィッティングである。多項式フィッティングの結果から見ると、ピークを過ぎた後はほぼ安定していることがわかる。ニューロン数と F-score の関係は、図 18 によって 7 次の多項式でフィッティングできる。また、多項式フィッティングから見ると、実際の結果と一致し、最も高い F-score はニューロン数 30 の近くに達する。

### 3.4 ネットワークロバスト性の検証

#### 3.4.1 異なる発火タイミング

スパイクソーティングアルゴリズムを成功させるには、高い性能が要求されるだけでなく、様々なデータに対してのロバスト性をテストする必要がある。

ロバスト性を検証するには、Bernert ら (2019) の論文によって、2 種類の特定のデータをテストする必要がある。断続的な発火活動と頻度が違う発火活動である。したがって、本研究では、先行研究の論文にある考えに基づき、彼らの考え方を模倣し、ロバスト性を検証するために検証実験を行った。

断続的な発火活動といえば、発火のタイミングが異なるニューロンに直面したとき、ニューラルネットワークがニューロンを正確に認識できるかどうかである。

解決したい問題は三つ：

- 1：このモデルのニューロンは、記憶があるのか。つまりスパイクが再び現すの時、新しいスパイクに識別されるか、それとも続いて学習できるのか（ニューロン 1 号）
- 2：一つの波形の学習できるの時間が、どのぐらいなのか（ニューロン 2 号）
- 3：学習の中で、新しいスパイクが現す場合、他今まで学習していないニューロンが学習できるのか（ニューロン 3 号）

学習過程で、ニューロン 1 号、2 号、3 号に対応する中間層のニューロンを見つければ、発火タイミングが異なるニューロンを状況に直面にしても、このモデルがロバストであることがわかる。

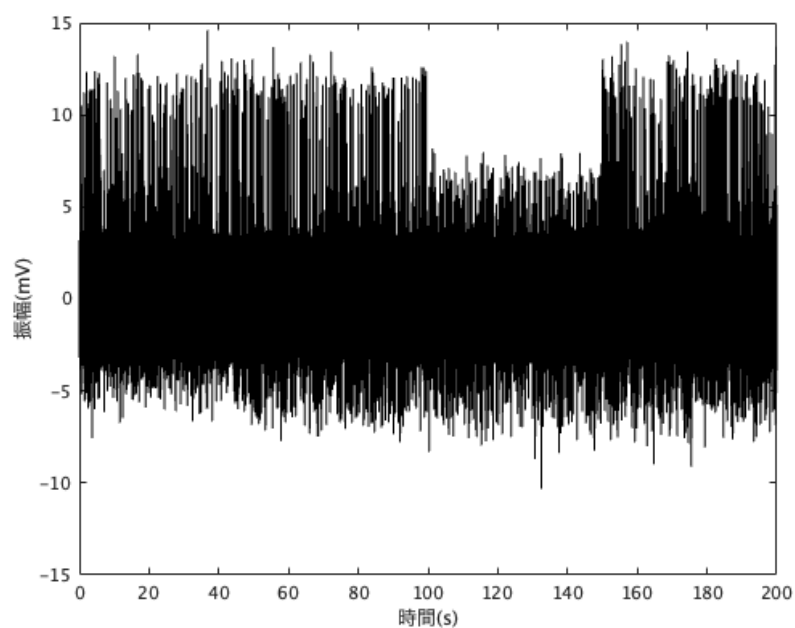


図 19: 発火タイミングが違う元スパイク信号

図 19 はタイミングロバスト性を検証するための元データである。発火のルールは、200 秒のうち、0-50 秒にニューロン 1,2 号、50 秒-100 秒にニューロン 2,3 号、100 秒-150 秒にニューロン 1,3 号は発火し、最後の 50 秒は再び全て発火するの発火ルールである。

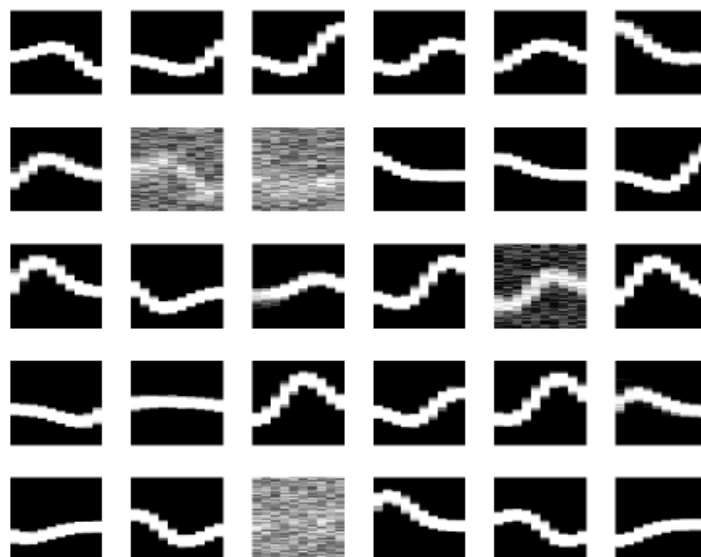


図 20: 発火タイミングが違う中間層重み

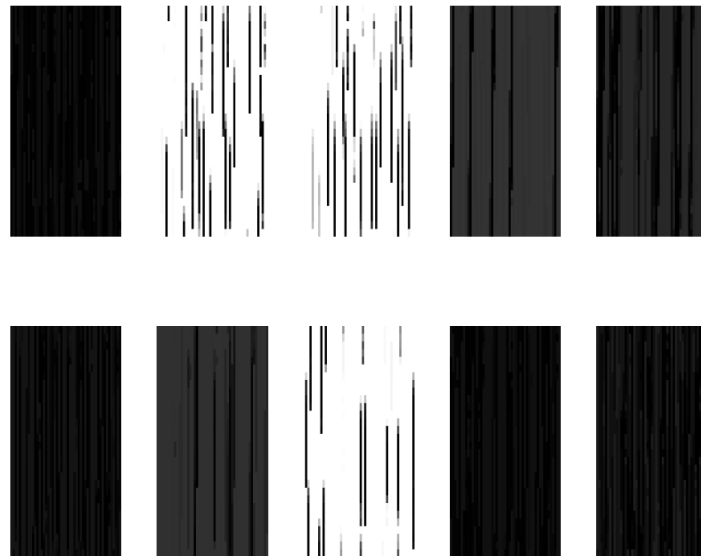


図 21: 発火タイミングが違う出力層重み

図 20,21 はそれぞれ中間層と出力層の重みである。各パネルが、一つの間層ニューロンあるいは出力層ニューロンに対応してる。スパイク波形は明らかに写し、興奮したニューロンも元データと同じ三つであることがわかる。したがって、ネットワークは正確に作動することがわかる。

図 22,23,24 はそれぞれニューロン 1 号、2 号、3 号に対応する中間層のニューロンの重み更新プロセスである。各パネル、10 秒ずつ中間層のニューロンの重みを可視化する。

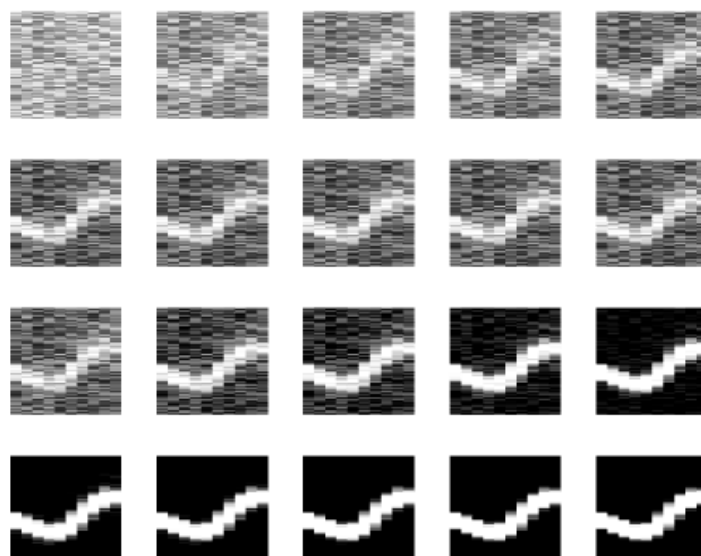


図 22: 発火タイミングが違うニューロン例:1

図 22 はニューロン 1 号を代表するの中間層ニューロンである。図 22 から見ると、例え一旦発火活動が消えたとしても、中間層ニューロンが学習の結果を記憶し、再び現す時が正確に学習続ける。

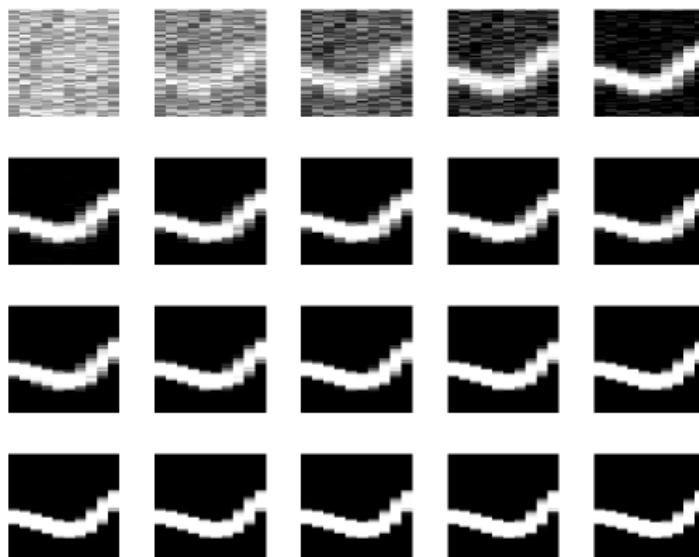


図 23: 発火タイミングが違うニューロン例:2

図 23 はニューロン 2 号を代表するの中間層ニューロンである。図 23 から見ると、一つの波形の学習できるの時間が 50 秒であり、つまり短い時間で波形を学習できる。

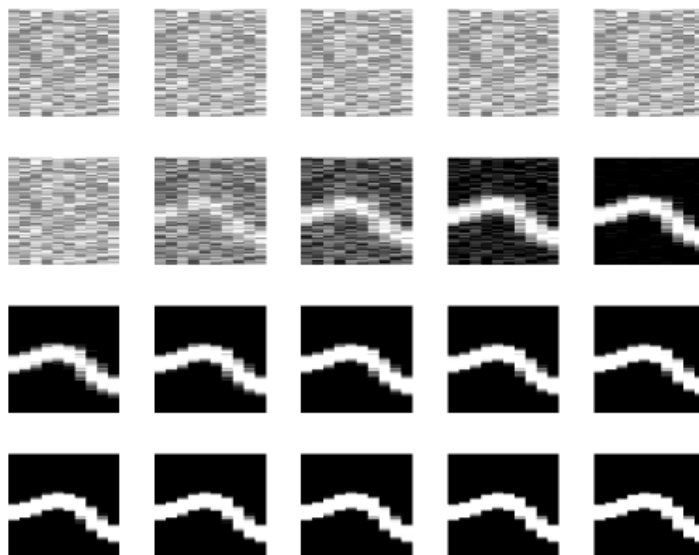


図 24: 発火タイミングが違うニューロン例:3

図 24 はニューロン 3 号を代表するの中間層ニューロンである。図 24 から見ると、最初がないの波形が現す時、

新しスパイクに識別されることがわかる。

三つのニューロン代表が全て見つけた上で、このモデルは、異なる発火タイミングに安定することが検証した。

### 3.4.2 異なる発火頻度

次は発火頻度が違う時の安定性検証である。

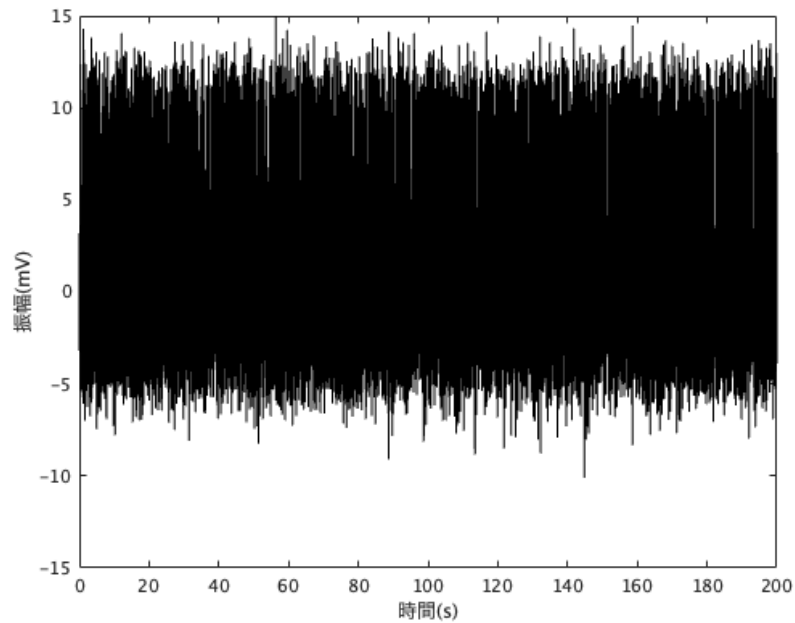


図 25: 発火頻度が違う元スパイク信号

図 25 は発火頻度が違うスパイクの元データである。検証条件として、三つニューロンの平均発火頻度、それぞれ 1,3,5Hz にする。

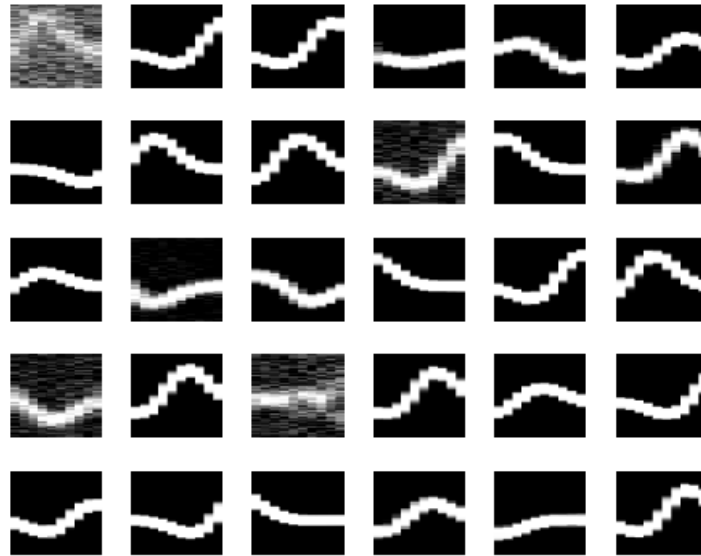


図 26: 発火頻度が違う中間層重み

図 26 は、発火頻度が違う時中間層の重みである。各パネルが、一つの間層ニューロンに対応してる。スパイク波形は明らかに写し、ネットワークは正確に作動することがわかる。また、結果として評価指標の F-score も 0.95 に達する。結果によって、このモデルは、発火頻度が違う時にも安定することが検証した。

## 4 考察

全体として、SNN ベースのニューラルネットワークはスパイクソーティングのタスクを実行できる。一方、SNN ニューラルネットワークはスパイクソーティングにおいてロバスト性を体現しており、スパイクデータの様々な状況に直面してもスパイクソーティングを解決することができる。同時に、実験中に多くの疑問が生じ、多くの新しい結論が得られた。

### 4.1 ニューロン数が Recall 率へ影響を与える原因

一つ目は、中間層のニューロン数を変えると、Recall 率に影響し、Precision 率にはほとんど影響しないことである。言い換えれば、ニューロンの数を変えても、ニューラルネットワークがより正確に認識できるようになるわけではなく、ニューロンの数を減らせば、学習のための生データからより多くのスパイクを検出できるようになる。この現象の理由を考えると、ニューロンの効率が関係していると思う。

ノイズレベルが高い場合、入力層とアテンションニューロンが識別した『スパイク』の中に、ノイズが入る可能性が高いである。中間層のニューロン数が余裕すぎて、使っていないニューロンがあるし、ノイズも学習した場合も可能である。出力層は、中間層が学習した結果に基づいて、結果を出すため、無駄なニューロンが入ると、時間内に学習の効率が低くなるかもしれないそれが原因で、効率が低いと、時間内で識別のスパイクが少なくなる。結果として、全てのスパイクが識別されていない、Recall 率が低い。

逆で、余裕という意味では、そのニューロン数だけで、データのスパイクを識別するのは十分である。効率が低いでも、全てのスパイクが識別した以上で、数が少ないでも、学習してのスパイクが正しい、その結果として、Precision 率にはほぼ影響がないであると私がそう思う。

全てのスパイクが識別されるのは、単位時間内で学習したの数が重要である。したがって、中間層のニューロン数に敏感であり、学習の効率が重要。Precision 率の方が、学習の効率が低いでも、式 (12) における Hit と Output が共に低くなって、結果としては影響がない。図 12 から見ると、ノイズが 1.75 の場合も Precision 率が 0.9 以上である。つまり Precision 率だけで、このモデルの性能が高い、SNN を用いたスパイクソーティングが可能である。

### 4.2 ニューロン数が性能へ影響を与える原因

次に議論したい問題点は、なぜ、ニューロン数を減少すると、性能が高くなるという疑問である。

例えそういう限界、最適な範囲があるとしても、一般的にどう決めるという疑問が現す。最後としても答えはないです。多項式フィッティングを試してみましたが、ピークを過ぎた後はほぼ安定していることがわかる。

このピークを見つけるには、勾配降下法などのさまざまなアルゴリズムを使用して極値を見つけることが考えられるかもしれないが、ニューロンの数と F-score の関係に対応する関数を知っていることが前提となり、おそらく実際の場合は、多項式フィッティングによる手を動かす探索する必要がある。

今回は非常に幸運で、全体的な極値の可能性のある 30 を見つけた。しかし、新しいデータの場合、極値の位置を判断するのは根拠はないし、局所的な極値を見つけた可能性もある。したがって、今の考え方では、どこが一番パフォーマンスが良いのかを判断するのは難しいと思う。アルゴリズムの考えについては、手を動かす探索しかできないと思う。

今回解けないが、研究の方向としては可能と思うし、リザーバーコンピューティングなどの研究方法も共通するはずである。一般に、SNN は NN 型ニューラルネットワークの制限を超えず、同じで段階的な構造を持ち、層内のニューロンも交流しない。しかし、もし最適なニューロン数が本当に存在するのであれば、SNN は構造に非常に敏感であるという特性を体現していると思う。この特性は STDP ルールや WTA ルールに関連しているのではないと思う。表面上は中間層ニューロンが削除されているが、実際には入力層から中間層、中間層から出力層に至る複数のシナプスが削除されている。情報伝達の効率は、全体ネットワークの性能に影響を与えると思う。

しかし、最適の条件を探すのは今の考えだと難しいであるが、興奮性ニューロンの割合によって、最適条件の判断を補助するのは可能である。図 27 から図 31 までは、それぞれ高いノイズレベル (1.75) の時、中間層ニューロン数が 25,26,30,34,35 のニューロン興奮状況である。結果としては、ニューロン数が 25 と 35 の時だけ、F-score が低い。これも図 17 と表 8 から確認できる。興奮ニューロン数の割合に注目すると、割合は以下表から確認できる。

表 9: 興奮ニューロンの割合

中間層ニューロン数 (個)	中間層興奮ニューロン数 (個)	割合 (%)
25	13	52.0
26	17	65.3
30	15	50.0
34	18	53.0
35	13	37.1

表 9 から見ると、ニューロン数が 25 と 35 の時だけ、興奮ニューロン数の絶対数値 (中間層興奮ニューロン数) と相対数値 (割合) が、どちらの一つが低い。これによって推測すると、ネットワークのパフォーマンスが、興奮ニューロンの絶対数値と相対数値に関連してるかもしれない。

興奮ニューロンの絶対数と相対数はともに重要で、それを実際探索のヒントになると私がそう思う。

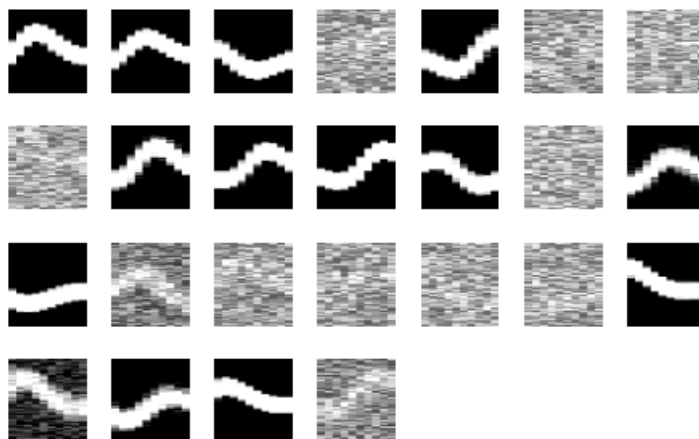


図 27: 中間層ニューロン数 25 での重み: ノイズレベル 1.75



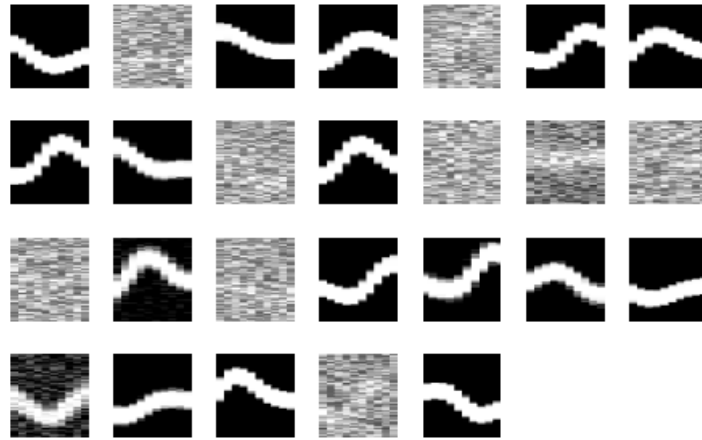


図 28: 中間層ニューロン数 26 での重み: ノイズレベル 1.75

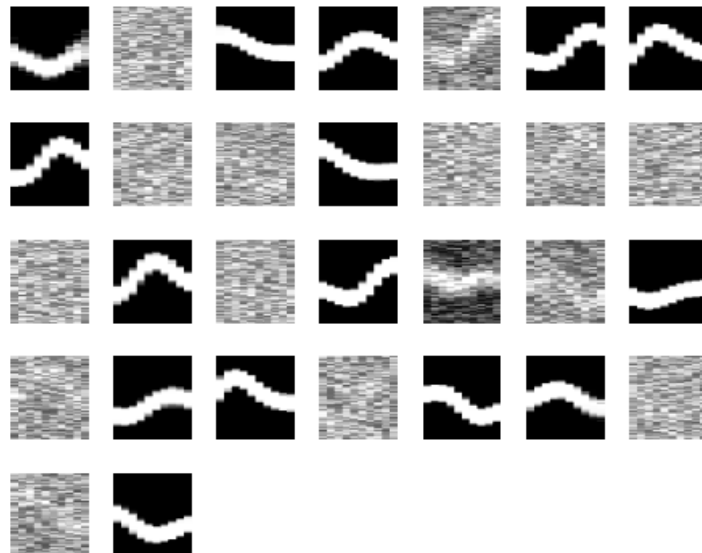


図 29: 中間層ニューロン数 30 での重み: ノイズレベル 1.75

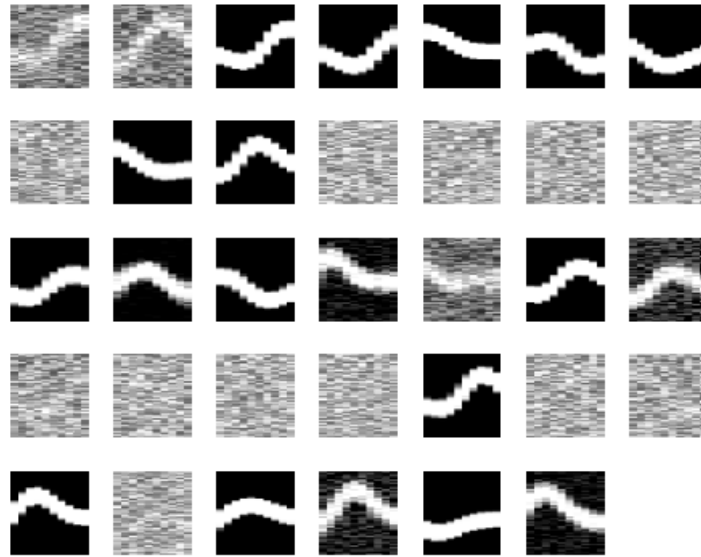


図 30: 中間層ニューロン数 34 での重み: ノイズレベル 1.75

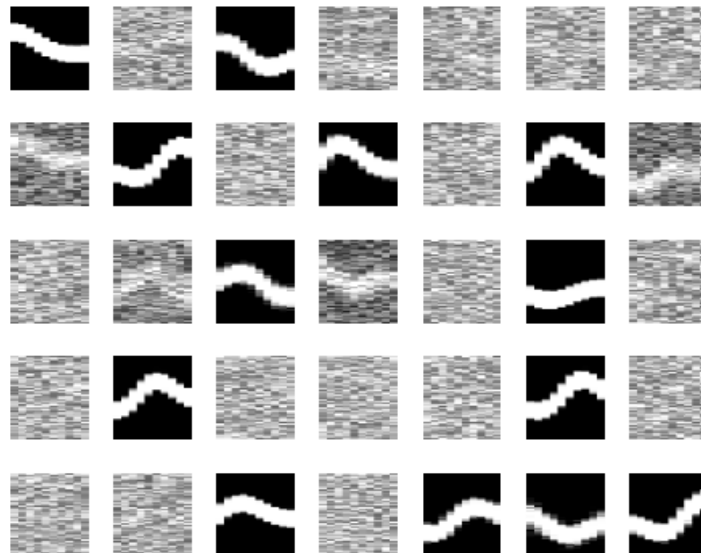


図 31: 中間層ニューロン数 35 での重み: ノイズレベル 1.75

### 4.3 SNN の将来性の考え

初期条件とし、Bernert ら (2019) によるり、中間層ニューロン数を 60 に設定する。Werner ら (2016) の結論によると、STDP の法則に従った各ニューロンの重みの変化は、約 200pJ のエネルギーを消費する。実験結果によると、アテンションニューロンは  $8 * 10^5$  回の発火を完成した。したがって、中間層ニューロンが重みを決まるとき、ニューラルネットワーク出力はおよそ  $9.6\mu\text{w}$  である。これは、SNN ニューラルネットワークの省エネ特性を十分に示している。今回研究に基づき、最適な性能を発揮するニューロン数は 30 であることがわかった上、本来の出力の半分を節約できるといえる。先に述べた出力層がスパイクソーティング後にリアルタイムでスパイク信号を出力できることや、今述べたような省エネルギー性を考慮すると、性能がどんどん向上しているの SNN は、デコード端末としてのブレインコンピュータインターフェイス (BCI) などの分野で活躍するの可能性もある。

## 5 謝辞

本論文執筆にあたり、適切な助言と丁寧なご指導をいただいた北野教授に心から深く感謝致します。テーマ選びに迷ったとき、熱心にアドバイスしてくださった北野先生のおかげさまで、自分の興味のあるトピックを選ぶことができ、研究を進めていく中でも、北野先生には毎回のゼミで研究の進捗状況についてアドバイスをいただいております。ご苦労大変感謝致します。

また、留学生として、いつも議論と精神的なサポートをしてくれた研究室のメンバーにも感謝したい。研究室のメンバーの熱意と親しみやすさにとても感謝しております。この研究を成功させることができたのは、両親の援助と立命館大学情報理工学部の授業料減免の経済的サポートのおかげさまで、無事に研究を終えることができました。皆様のサポートを本当に感謝致します。

## 参考文献

- [1] Buccino, Alessio P., Samuel Garcia, and Pierre Yger. "Spike sorting: new trends and challenges of the era of high-density probes." *Progress in Biomedical Engineering* 4.2 (2022): 022005.
- [2] Zhang, Yiwei, et al. "A robust spike sorting method based on the joint optimization of linear discrimination analysis and density peaks." *Scientific Reports* 12.1 (2022): 15504.
- [3] Toosi, Ramin, Mohammad Ali Akhaee, and Mohammad-Reza A. Dehaqani. "An automatic spike sorting algorithm based on adaptive spike detection and a mixture of skew-t distributions." *Scientific Reports* 11.1 (2021): 13925.
- [4] Saunders, Daniel J., Hava T. Siegelmann, and Robert Kozma. "Sdp learning of image patches with convolutional spiking neural networks." *2018 international joint conference on neural networks (IJCNN)*. IEEE, 2018.
- [5] Diehl, Peter U., and Matthew Cook. "Unsupervised learning of digit recognition using spike-timing-dependent plasticity." *Frontiers in computational neuroscience* 9 (2015): 99.
- [6] Bernert, Marie, and Blaise Yvert. "An attention-based spiking neural network for unsupervised spike-sorting." *International journal of neural systems* 29.08 (2019): 1850059.
- [7] Rokai, János, et al. "Elvisort: encoding latent variables for instant sorting, an artificial intelligence-based end-to-end solution." *Journal of Neural Engineering* 18.4 (2021): 046033.
- [8] Lecoq, Jérôme, et al. "Removing independent noise in systems neuroscience data using DeepInterpolation." *Nature methods* 18.11 (2021): 1401-1408.
- [9] Wu, Tong, et al. "Learning to sort: few-shot spike sorting with adversarial representation learning." *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019.
- [10] Werner, Thilo, et al. "Spiking neural networks based on OxRAM synapses for real-time unsupervised spike sorting." *Frontiers in neuroscience* 10 (2016): 474.
- [11] Masquelier, Timothée, Rudy Guyonneau, and Simon J. Thorpe. "Competitive STDP-based spike pattern learning." *Neural computation* 21.5 (2009): 1259-1276.
- [12] Bichler, Olivier, et al. "Visual pattern extraction using energy-efficient "2-PCM synapse" neuromorphic architecture." *IEEE Transactions on Electron Devices* 59.8 (2012): 2206-2214.
- [13] Adamos, Dimitrios A., Efstratios K. Kosmidis, and George Theophilidis. "Performance evaluation of PCA-based spike sorting algorithms." *Computer methods and programs in biomedicine* 91.3 (2008): 232-244.