

# HTM を用いた脳波異常検知

令和3年度 修士論文

横浜市立大学大学院

生命ナノシステム科学研究科

物質システム科学専攻

土屋 慶典

## 目次

1.章 序論.....	8
1.1 研究背景.....	8
1.2 研究目的.....	9
2.章 Literature review (関連研究).....	10
2.1 近年使用されている人工知能.....	10
2.2 HTM の特徴.....	11
2.3 HTM と深層学習の違い.....	13
2.4 脳波.....	14
3.章 HTM.....	16
3.1 SDR.....	16
3.2 学習アルゴリズム.....	17
3.3 Anomaly detection.....	24
3.4 NuPIC.....	27
4.章 実験概要.....	30
4.1 実験方法.....	30
5.章 結果と考察.....	32
6.章 まとめ.....	35
7.章 謝辞.....	36
8.章 文献.....	37
付録.....	38

# HTM を用いた脳波異常検知

土屋 慶典

指導教員 Micheletto Ruggero

副指導教員 野々瀬 真司

副指導教員 谷本 博一

## HTM を用いた脳波異常検知

物質システム科学専攻 土屋慶典  
指導教員 Micheletto Ruggero

### 【重要語句】

**HTM**：人間の脳の機能を模倣したニューロン的一种。主に大脳新皮質を模倣している。

**SDR**：HTMにおいて情報を処理する際に用いられる表現。実際に人間の脳の神経細胞の働きにヒントを得ており、0, 1で表されるビットの内1 (ON ビット) となる割合は2%以下となっている。

**ヘブの法則**：HTMにおいて学習アルゴリズムとして用いられる。各シナプスの接続値を増減させることで、シナプス接続の可否を変化させる。あるセルの発火の後に他のセルが発火した時にその2つのセル間のシナプス結合を強化させる。

**Anomaly detection**：本論文内では、HTMが学習を通じて、入力データが異常であるかを判断するものと定義する。Anomaly likelihood という値を算出することで異常な入力であるかを判断する。

**Anomaly likelihood**：入力データが異常であるかを判断する数値。値が高いほど入力データが異常であることを表す。HTMの学習によって算出される。

**アーチファクト**：実験で生じた脳波以外のノイズ。

### 【研究背景と目的】

人工知能の研究は近年急速に発展しており、私達の生活に密接に関わりを持つようになった。しかし、身近で使われている人工知能には、特定の課題しか解くことが出来ない、汎用性に欠けるといった課題がある。そこで私はHTM(Hierarchical temporal memory)という人工知能に着目した。HTMは生物学的、神経学的観点からアプローチをとっており、人間の脳の働きに近い。したがって、将来的に上記の課題を解決する汎用的人工知能として期待されている。HTMには、ノイズに対する堅牢性が高い、時系列データを取り扱う、異常検知に強い、といった特徴があるため、異常脳波の検知にHTMを応用する可能性を探った。脳波検知は脳波に大量のノイズが含まれる、人間の目で検知するには長年の経験と知恵が必要、といった問題を抱えており、脳波検知にHTMが応用できれば、医師の負担軽減、てんかんなどの病気の早期発見に役立つ。実験で得られた異常脳波をHTMがどの程度検知できるか定量的に調べることを本研究の目的とした。本研究では、numenta社が開発したソフトパッケージNuPICの皮質学習アルゴリズム(CLA)と呼ばれるHTMの公式実装を用いた。

## Anomaly detection using HTM

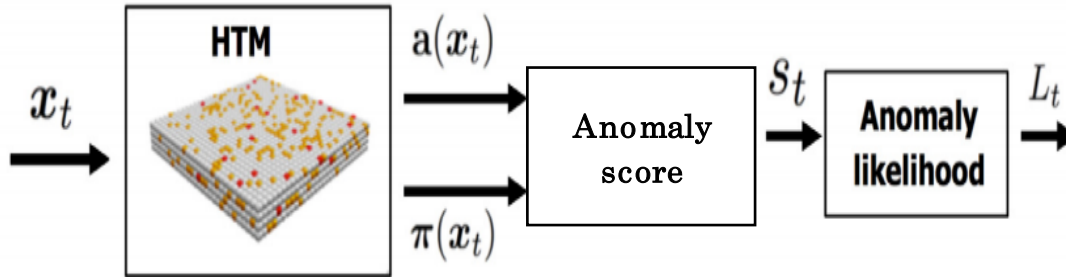


図 1 anomaly detection 概要図

出典 Ahmad, Subutai, Lavin, Alexander, Purdy, Scott, Agha, Zuha. Unsupervised real-time anomaly detection for streaming data. *Nuerocomputing*, 262, p134-137, 2017.8.24

### 【実験】

本実験では、刺激によって生じた脳波を異常脳波と定義し、眼球運動、音刺激、光刺激による 3 パターンの実験を行った。アーチファクト対策として、閉眼、安静、暗室の環境下で測定をした。1 つ目の実験では、被験者による瞬きを眼球運動とし、4 分間の測定時間で適当なタイミングで数回瞬きを促した。2, 3 つ目の実験では、3 分間の測定時間で最後の 1 分間に光、音刺激を加えた。脳波測定はフューテックエレクトロニクス株式会社性 BrainPro (ブレインプロ) FM-939 及び PC 用ソフトウェア Analyzer+ (アナライザープラス) FS-AP により行った。BrainPro で検出した脳波を Analyzer+ は 1024Hz で原脳波を取得し、高速フーリエ変換 (FFT、窓関数: 矩形) により、1.0Hz~30.0Hz まで 0.5Hz 毎にパワースペクトル解析する。電極は 10/20 法による FP2 (及び FP1、A1) に装着した。瞬きによる眼球運動によって生じたアーチファクト、音、光刺激によって生じた脳波の変化を異常脳波とし、HTM で学習させることによって異常脳波を検出することを実験目的とした。通常、音、光や痛みなどの刺激に対して脳波は反応を示すため、反応性がない場合は脳に何らかの異常がある可能性がある。



図 2 実験風景

出典 <https://www.futek.co.jp/products/humantech/fm-939/images/fm-939-scene1.jpg>  
 (2021-07-24 最終ログイン)

【結果と考察】

各結果の上部分は測定した脳波、下部分は anomaly likelihood を表す。それぞれの最初の段階で見られる、脳波が安定していない箇所は被験者の筋肉の動きによって生じたアーチファクトである。学習が進んでいない段階では HTM が入力されたデータのほとんどを異常と判断し、anomaly likelihood の値が高くなっているが、時間が経過するにつれある程度安定する。図 2 の中盤以降に見られる 2 つの矢印は眼球運動によって生じたアーチファクトである。眼球運動による EEG の値の変化に応じて、anomaly likelihood の値も上昇している。これは HTM が異常脳波（眼球運動によって生じたアーチファクト）を検出出来ていることが分かる。しかし、学習が進んだにも関わらず、anomaly likelihood の値は 0 に収束せず、常に何らかの異常入力（異常脳波）があった可能性を表す。これは HTM が様々な要因によって発生したアーチファクトに異常を検出していることが考えられる。

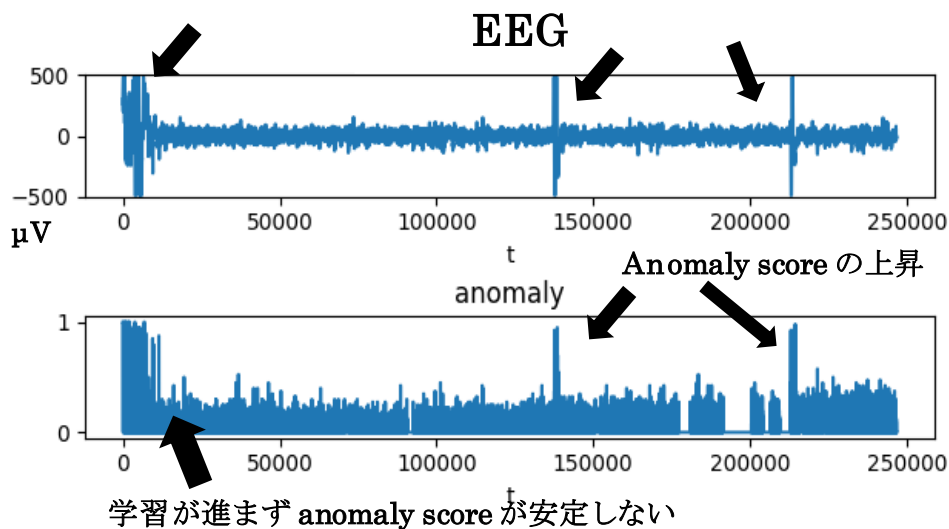


図 3 結果①

図 5-2、5-3は図 5-1 と同様に序盤の段階は学習が進んでいないことに加え、アーチファクトが多く含まれているため、anomaly likelihood は安定していない。しかし、学習がある程度進み、刺激を加えた後も anomaly likelihood の値は大きく推移しなかった。anomaly likelihood が推移しなかった原因として、そもそも刺激に対して脳波の変化がそこまでなかった、脳波の変化が少なく、アーチファクトによる脳波変化と大差がなかったことが挙げられる。今後は、公開データなどを活用し、精度を上げていきたい。また、異なる人工知能や、分析手法と精度を比較したい。

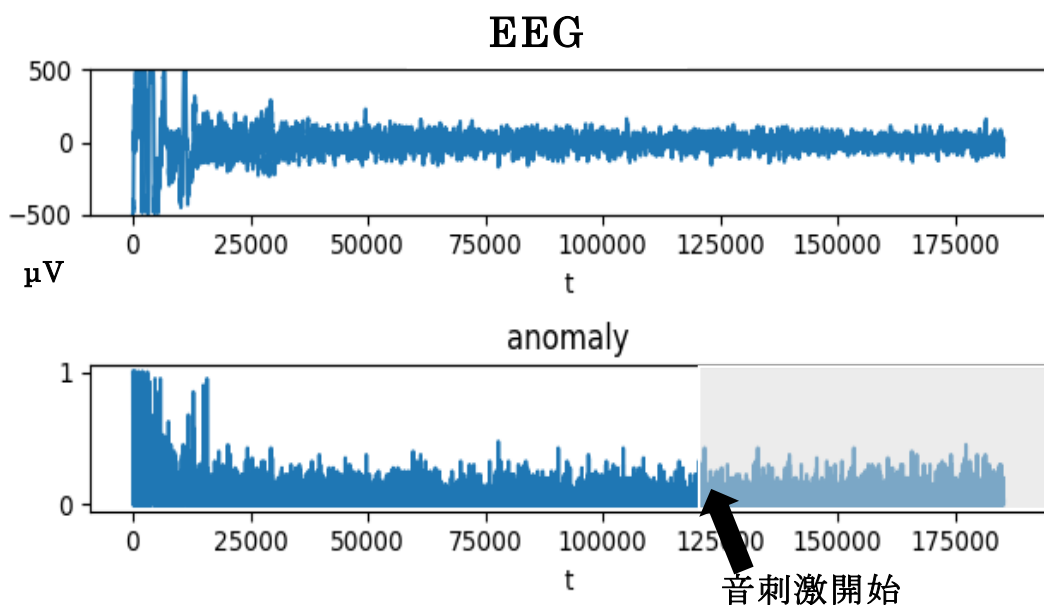


図 4 結果②

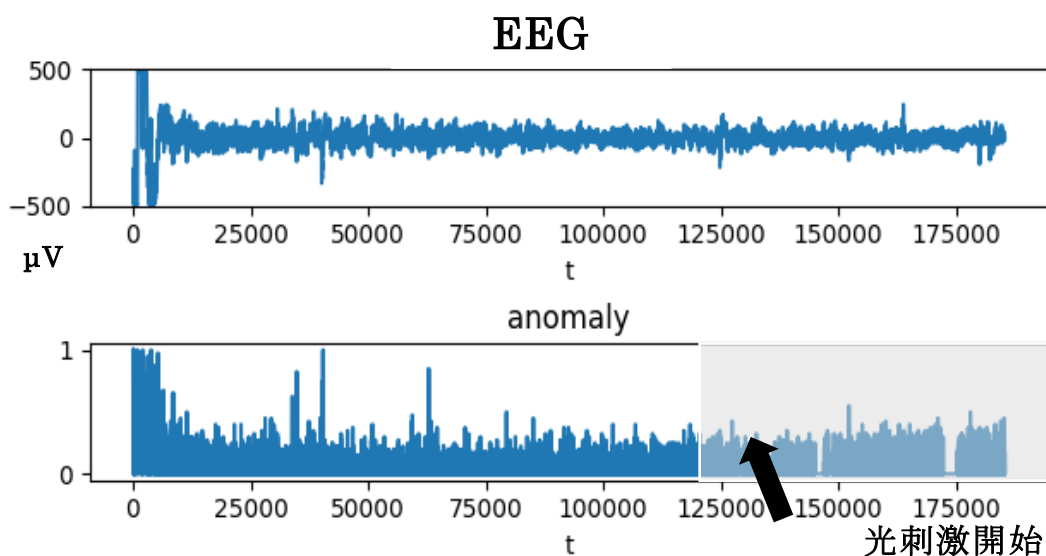


図 5 結果③

## 1.章 序論

### 1.1 研究背景

近年、人工知能は私達の生活に欠かせない存在になっており、図 1-1からもわかる通り、注目度は年々高まっている。特に、深層学習（deep learning）は、様々な場面に応用されている。しかし、そうした人工知能にも特定の課題しか扱うことが出来ない、汎用性に欠けるといった課題がある。こうした課題を解決する人工知能として、HTM（Hierarchical temporal memory）が挙げられる。HTM は他の人工知能と比較して、生物学的、神経学的観点からアプローチをとっており、人間の脳の働きに近い。したがって、将来的には汎用的人工知能として期待されている。しかし、HTM に関する研究の数は少なく、明らかにされていない点が多々ある。そこで、HTM を現在研究が進められていない分野に応用する可能性を探るために研究を始めた。

NUMBER of PEER-REVIEWED AI PUBLICATIONS, 2000-19

Source: Elsevier/Scopus, 2020 | Chart: 2021 AI Index Report

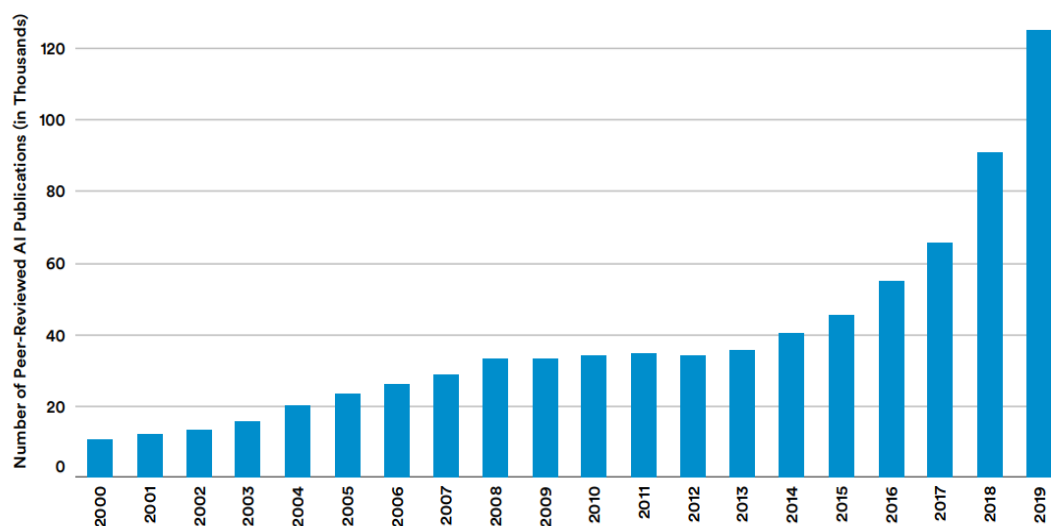


図 1-1 査読付き AI 論文数の推移

出典

<https://aiindex.stanford.edu/wp-content/uploads/2021/11/2021-AI-Index-Report-Master.pdf>

2022年1月27日最終ログイン



## 1.2 研究目的

HTM はノイズに対する堅牢性が高い、リアルタイムのデータに対する異常検知の精度が高い特徴を持つ。これらの特徴は脳波分析との親和性が高い。脳波は非常に小さな電位変化をすることに加え、脳波を直接測定することが難しいため、ノイズによる影響を受けやすい。こうした問題は HTM を活用することによって解決する可能性を秘めている。そこで、実験で脳波を測定し、HTM に学習させることで異常脳波を検出することを研究目的とした。HTM を脳波分析に活用することが出来れば、てんかんなどの病気の治療や、ニューロフィードバック、BCI などの分野に活用できる。本研究では、numenta 社が開発したソフトウェア NuPIC に含まれる皮質学習アルゴリズム (CLA) と呼ばれる HTM の公式実装を用いた。

## 2.章 Literature review (関連研究)

### 2.1 近年使用されている人工知能

ここ数年で起きている人工知能の大きなブレークスルーは **deep learning** (深層学習) に大きく起因している。特に画像認識、音声認識、自然言語処理、ゲームなどの領域に応用されており、人間の能力を上回る人工知能も多々存在する。深層学習は深い層を重ねることで学習精度を上げるように工夫したニューラルネットワークを用いた機械学習の一種である。図は深層学習の大まかな形を表したものである。図 2-1 のように中間層を多層にすることで情報伝達と処理を増やし、特徴量の精度や汎用性をあげたり、予測精度を向上させたりすることが可能となる。深層学習を応用した分野に空間的構造をモデル化した、畳み込みニューラルネットワーク、時間的構造をモデル化したリカレントニューラルネットワークなどが挙げられる。しかし、学習用のデータが限られている時に上手く機能しない、データのパターンが常に変化している問題には対応できないなどの課題があり、主に達成すべき課題が明確に定義されているものに応用されている。

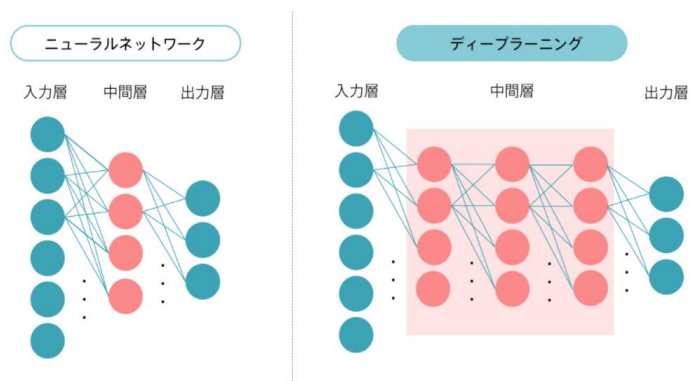


図 2-1 ニューラルネットワークと深層学習の違い

出典

<https://leapmind.io/blog/2017/06/16/%E3%83%87%E3%82%A3%E3%83%BC%E3%83%97%E3%83%A9%E3%83%>

2022年1月21日最終ログイン

## 2.2HTM の特徴

HTM は人間の脳の機能を模倣したニューロン的一种であり、特に大脳新皮質を模倣している。図 2-2 の 3 種類のニューロンを用いて、HTM の大まかなニューロン構造を説明する。図 2-2 の左図は単純な人工ニューロンである。基本的な人工ニューロンはシナプスが比較的少なく、樹状突起がなく、シナプスの重みを変化させることで学習を行う。入力加重和を計算し、出力値を決定する。図 2-2 の真ん中の図は大脳新皮質で最も一般的なタイプである錯体型ニューロンである。生物学的なニューロンは樹状突起に沿って配置された何千ものシナプスを持つ。樹状突起はニューロンが膨大な量のパターンを認識させるための処理要素となる。新しいシナプスの形成、使われなシナプスを除去することで学習をする。また、複数のシナプスを結ぶ領域を持ち、これは他のニューロンにも影響を与える。図 2-2 の右図は HTM の人工ニューロンである。図 2-2 の真ん中の図と同様に、アクティブな樹状突起上に配置された数千のシナプスを持ち、多くのパターンを認識する。2 値のシナプスを用い、新しいシナプスの強化、未使用のシナプスを抑制させることをモデル化することで学習する。HTM ニューロンは生物学的なニューロンの全てをモデル化するのはではなく、新皮質の情報理論的に不可欠なものだけをモデル化している。

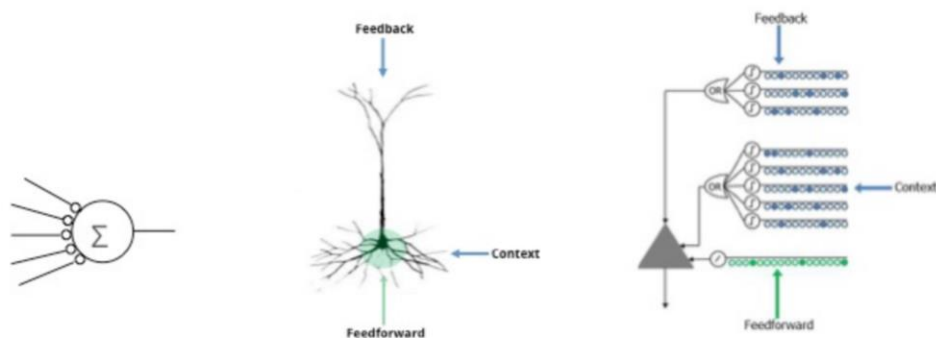


図 2-2 3 種類のニューロン

出典

<https://numenta.com/assets/pdf/biological-and-machine-intelligence/BaMI-HTM-Overview.pdf>

2022 年 1 月 21 日最終ログイン

主な特徴としては以下のものが挙げられる。

- 階層構造を持つ

- 疎分散表現 (SDR)
- 堅牢性が高い
- オンライン学習
- シーケンス学習
- 異常検知の分野に強い

HTM 理論は非常に複雑であり、本質的に理解を深めることは難しいが、3章で HTM のより詳しい内容を本論文に関わる部分を中心に説明する。

## 2.3 HTM と深層学習の違い

表 2-3 は HTM と深層学習の相違点を表したものである。まず初めに HTM は生物学的観点からのアプローチをとっており、人間の脳の働きに近い性質を持つのに対し、深層学習は数学的、工学的アプローチをとっている。また、オンライン学習、教師無し学習、学習にヘブの法則を用いるといった点は深層学習に比べ、より人間の脳の働きに即している。現段階では深層学習は特定の課題を扱う際に非常に強みを持ち、我々の生活への浸透度は高いが、HTM は将来的に深層学習では難しい領域である汎用的人工知能の分野に応用できる可能性を持つ。

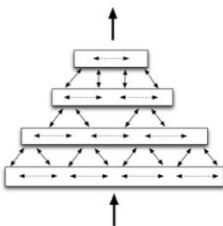
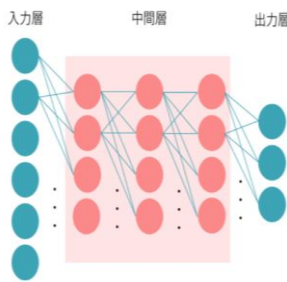
	HTM	深層学習
		
アプローチ	生物学的	数学的、工学的
学習メカニズム	ヘブの法則	誤差逆伝播
学習タイプ	教師なし	教師あり

表 2-1 HTM と深層学習の違い

出典

<https://numenta.com/assets/pdf/whitepapers/hierarchical-temporal-memory-cortical-learning-algorithm-0.2.1-en.pdf>

2022 年 1 月 21 日最終ログイン

出典

<https://leapmind.io/blog/2017/06/16/%E3%83%87%E3%82%A3%E3%83%BC%E3%83%97%E3%83%A9%E3%83%>

2022 年 1 月 21 日最終ログイン

## 2.4 脳波

脳波（electroencephalogram：EEG）とは、脳神経細胞の活動に伴って生じる電位変化を多くの場合体表から記録したもので、脳波計は高感度の増幅器であり、数十  $\mu\text{V}$  の微小な電位変化を増幅して数 mm 程度の振れとして記録する[15]。脳波スペクトルには帯域ごとに名称とその周波数帯を示しており、研究者によって多少の相違はあるが、表のように分類される。

脳波の種類（名称）	主な周波数範囲
$\delta$ 波	1～4Hz
$\theta$ 波	4～8 Hz
$\mu$ 波	7～13 Hz
$\alpha$ 波	8～13 Hz
$\beta$ 波	14～30 Hz
$\gamma$ 波	30 Hz～

表 1 脳波の分類

脳波は微小な電位変化をするため、脳以外から発生する電位が混入しやすい。こうした脳波以外のノイズをアーチファクトと表す。アーチファクトには下記のような種類が存在する。

被験者に起因するアーチファクト

- 筋電図
- 心電図
- 眼球運動、瞬き
- 発汗
- 呼吸
- 体動
- 不随意運動

被験者以外に起因するアーチファクト

- 交流障害（電磁誘導、静電誘導、漏れ電流）
- 記録装置（脳波計）
- 環境的因子（被験者付近での人の動き）
- 被験者に装着されたアーチファクト源（金属義歯）

このようなアーチファクトによる影響を最小源に抑えることは脳波を測定する際には非常に重要となる。ただ、全てのアーチファクトを取り除くことは現実的に難しい。

### 3.章 HTM

#### 3.1SDR

本章では HTM の詳細を項目ごとに説明する。HTMではデータを表す基礎として、SDR (sparse distributed representations) を用いる。SDR は図 3-1 のように 1 または ON ビットが非常に少なく、長い 2 値ベクトル表現であり、入力データが圧縮された意味的表現を構成する。基本的に ON ビットとなる割合は 2%以下に設定される。実際に人間の脳の神経細胞の働きは 2%以下であり、人間の脳の特徴を模している。入力されたデータが SDR の形にエンコードされた後に HTM 内で計算される。SDR を用いることで、多くの情報を保持したまま、高い堅牢性を保つことが出来る。また、複数の情報を格納することが可能なユニオン特性を持つため、同時に複数の予測が可能である[12]。

1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0

図 3-1 SDR (疎分散表現)



### 3.2 学習アルゴリズム

HTMは、セルと名付けられたノードを2次元マップ上に配置したものとなっている。複数のセルから成り立つカラムの集合が図 3-2 のように構成される。カラムの組み合わせから、1つのパターンを表現する。時間が進むごとにこのパターンが変化することで、時系列データを表現する。各セルはそれぞれ活性化状態、非活性化状態、予測状態と遷移する。また、各セルはセグメント構造を持っており、大きく分けて、入力セグメントとセグメント集合の2つからなる。入力セグメントは入力を受け取り、各セルに1つある。セグメント集合も各セルに1つあり、接続セグメントを複数持つ。



図 3-2 HTM の構成

出典 [16]

学習アルゴリズムはセル間の繋がりであるシナプス接続の可否を変化させるものである。シナプス接続の可否は各シナプスの接続値を増減させることで変化させる。これはあるセルの発火の後に他のセルが発火した時にその2つのセル間のシナプス接続を強化させるというものである。この学習則は成人の脳皮質が感覚活動に応じて新しいシナプスを急速に生成することを示した、活動依存のシナプス形成に関する神経科学の研究にヒントを得ており、ヘブ則に即している。極めて単純に表すと、シナプスの両側（入力と出力）のニューロンが高い相関を持った出力をするとき、2つのニューロン間のシナプスは強化されるというものだ。図 3-3 は HTM を構成しているセルの状態が変化することによって学習をした際の概要図である。

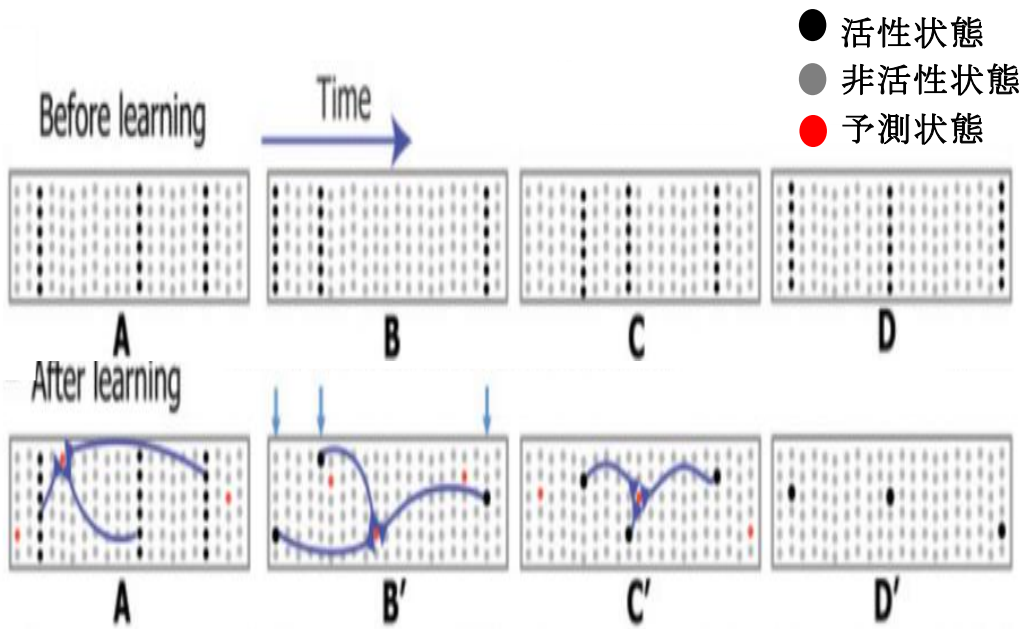


図 3-3 HTM のアルゴリズム  
出典[12]

ここで上記の学習アルゴリズムを数式を用いて説明する。まず初めにパラメータを設定する。HTM のカラムの数を  $N$ 、1 カラム中のセルの数を  $M$  とする。この時、セルの総数は  $M \times N$  となる。時刻  $t$  における活性化状態のセルの集合を  $M \times N$  の 2 進行列  $A^t$ 、 $a_{ij}^t$  と表す。 $a_{ij}^t$  は活性化状態の  $i$  番目のカラムにある、 $j$  番目のセルである。同様に時刻  $t$  における予測状態のセルの集合を  $M \times N$  の 2 進行列  $\Pi^t$ 、 $\pi_{ij}^t$  と表す。 $\pi_{ij}^t$  は予測状態の  $i$  番目のカラムにある、 $j$  番目のセルである。各セルにあるセグメント集合の中で  $d$  番目のセグメントの  $j$  番目のカラムにある  $i$  番目のセルを  $D_{ij}^d$  とする。またセグメント集合中のセグメントはそれぞれ各セルから他のセルへの接続値を保持しているため、セグメントの接続値がしきい値を超えたセルのみを記録した 2 進行列  $\tilde{D}_{ij}^d$  接続の可否のみを表す。入力セグメントに入力を受けたセルが存在するカラムの集合を勝者カラムとし、 $W^t$  とする。活性化状態の計算は図 3-4 式で表される。

$$a_{ij}^t = \begin{cases} 1 & \text{if } j \in W^t \text{ and } \pi_{ij}^{t-1} = 1 \\ 1 & \text{if } j \in W^t \text{ and } \sum_i \pi_{ij}^{t-1} = 0 \\ 0 & \text{otherwise} \end{cases}$$

図 3-4 活性化状態の計算

図 3-4 式から 2 つの条件下でセルが活性化状態になることが分かる。1 つ目はセルが勝者カラム中のカラムに含まれており、そのセルが予測状態に遷移している場合である。2 つ目はセルが勝者カラム中のカラムに含まれており、そのセルを含むカラム中に予測状態に遷移しているセルが 1 つも存在しない場合である。予測状態の計算は図 3-5 式で表される。

$$\pi_{ij}^t = \begin{cases} 1 & \text{if } \exists_d \|\tilde{\mathbf{D}}_{ij}^d \circ \mathbf{A}^t\|_1 > \theta \\ 0 & \text{otherwise} \end{cases}$$

図 3-5 予測状態の計算

$\theta$  は接続しているシナプス接続の数における閾値である。この閾値を超えている時にセルとセルが接続しているとみなす。セグメント集合中のある接続セグメントに関して活性化状態にあるセルとその接続セグメントによって接続しているセルが予測状態に遷移する。次に接続値の更新の説明をする。以下 3 つの場合にセグメント集合が持つ接続値を更新する。

① セルが予測状態になり、その後活性化状態に遷移した場合 (図 3-6)。

$$j \in \mathbf{W}^t \quad \text{and} \quad \exists_i (\pi_{ij}^{t-1}) > 0$$

図 3-6 接続値の更新①

② 勝者カラム中にあるどのセルも予測状態になっていなかった場合 (図 3-7)。

$$j \in \mathbf{W}^t \quad \text{and} \quad \forall_i (\pi_{ij}^{t-1}) = 0$$

図 3-7 接続値の更新②

③ セルが予測状態になっていたが、その後活性化状態に遷移しなかった場合 (図 3-8)。

$$j \notin \mathbf{W}^t \quad \text{and} \quad \exists_i (\pi_{ij}^{t-1}) > 0$$

図 3-8 接続値の更新③

ここで①と②の場合はシナプス接続を強化する。したがって、セグメント集合中にある接続値を増加させる。①の場合はセグメント集合の中ですでに接続値が閾値を超えている接続セグメントに関して、②の場合はセグメント集合のなかで一番大きい接続値を持つ接続セグメントに関して図 3-9 の更新式を適用する。

$$\Delta \mathbf{D}_{ij}^d = p^+ \dot{\mathbf{D}}_{ij}^d \circ \mathbf{A}^{t-1} - p^- \dot{\mathbf{D}}_{ij}^d \circ (\mathbf{1} - \mathbf{A}^{t-1})$$

図 3-9 更新式①

$p^+$ 、 $p^-$ は学習率であり、 $p^+ > p^-$ 。  $\dot{\mathbf{D}}_{ij}^d$ は $\mathbf{D}_{ij}^d$ の正の値をもつ位置のみを取り出した2進行列である。  $\circ$ は要素積を表し、行列の要素同士の積をとる演算である。図 3-10 の更新式は不活性シナプスの接続値を減少させ、活性シナプスの接続値を増加させることによってシナプス接続を強化する。

$$\dot{\mathbf{D}}_{ij}^d = \begin{cases} 1 & \text{if } \mathbf{D}_{ij}^d > 0 \\ 0 & \text{otherwise} \end{cases}$$

図 3-10 更新式②

また、③の場合はセグメント集合の中で閾値を超えている接続値を減少させる。これはシナプス接続の減衰を再現している (図 3-11) 。  $p^{--}$ 、 $p^-$ は学習率であり、 $p^{--} > p^-$

$$\begin{aligned} \Delta \mathbf{D}_{ij}^d &= p^{--} \dot{\mathbf{D}}_{ij}^d \\ \text{where } a_{ij}^t &= 0 \text{ and } \|\tilde{\mathbf{D}}_{ij}^d \circ \mathbf{A}^{t-1}\|_1 > \theta, \\ \text{where } p^{--} &\ll p^- \end{aligned}$$

図 3-11 更新式③

最後にこれらの学習アルゴリズムを用いて、実装された `numenta` による疑似コードを添付する。 `sp`、`tm algorithm` の2つのステップに分かれて学習が進む[12]。この疑似コードは多くの専門用語を含んでいるため、詳しい理解を深めるためには[6]を参照する必要がある

る。このコードを理解することによって、より体系的な学習アルゴリズムの理解が可能となる。

```
1. for c in columns
2.     overlap(c) = 0
3.     for s in connectedSynapses(c)
4.         overlap(c) = overlap(c) + input(t, s.sourceInput)
5.     overlap(c) = overlap(c) * boost(c)
6. for c in columns
7.     minLocalActivity = kthScore(neighbors(c), numActiveColumnsPerInhArea)
8.     if overlap(c) > stimulusThreshold and
9.         overlap(c) ≥ minLocalActivity then
10.        activeColumns(t).append(c)
11. for c in activeColumns(t)
12.     for s in potentialSynapses(c)
13.         if active(s) then
14.             s.permanence += synPermActiveInc
15.             s.permanence = min(1.0, s.permanence)
16.         else
17.             s.permanence -= synPermInactiveDec
18.             s.permanence = max(0.0, s.permanence)
19. for c in columns:
20.     activeDutyCycle(c) = updateActiveDutyCycle(c)
21.     activeDutyCycleNeighbors = mean(activeDutyCycle(neighbors(c)))
22.     boost(c) = boostFunction(activeDutyCycle(c), activeDutyCycleNeighbors)
23.     overlapDutyCycle(c) = updateOverlapDutyCycle(c)
24.     if overlapDutyCycle(c) < minDutyCycle(c) then
25.         increasePermanences(c, 0.1*connectedPerm)
26. inhibitionRadius = averageReceptiveFieldSize()
```

図 3-1 2 sp 疑似コード  
出典[6]

```

1. for column in columns
2.   if column in activeColumns(t) then
3.     if count(segmentsForColumn(column, activeSegments(t-1))) > 0 then
4.       activatePredictedColumn(column)
5.     else
6.       burstColumn(column)
7.   else
8.     if count(segmentsForColumn(column, matchingSegments(t-1))) > 0 then
9.       punishPredictedColumn(column)
10.  function activatePredictedColumn(column)
11.    for segment in segmentsForColumn(column, activeSegments(t-1))
12.      activeCells(t).add(segment.cell)
13.      winnerCells(t).add(segment.cell)
14.
15.    if LEARNING_ENABLED:
16.      for synapse in segment.synapses
17.        if synapse.presynapticCell in activeCells(t-1) then
18.          synapse.permanence += PERMANENCE_INCREMENT
19.        else
20.          synapse.permanence -= PERMANENCE_DECREMENT
21.
22.      newSynapseCount = (SYNAPSE_SAMPLE_SIZE -
23. numActivePotentialSynapses(t-1, segment))
24.      growSynapses(segment, newSynapseCount)
25.  function burstColumn(column)
26.    for cell in column.cells
27.      activeCells(t).add(cell)
28.
29.    if segmentsForColumn(column, matchingSegments(t-1)).length > 0 then
30.      learningSegment = bestMatchingSegment(column)
31.      winnerCell = learningSegment.cell
32.    else
33.      winnerCell = leastUsedCell(column)
34.      if LEARNING_ENABLED:
35.        learningSegment = growNewSegment(winnerCell)
36.
37.    winnerCells(t).add(winnerCell)
38.
39.    if LEARNING_ENABLED:
40.      for synapse in learningSegment.synapses
41.        if synapse.presynapticCell in activeCells(t-1) then
42.          synapse.permanence += PERMANENCE_INCREMENT
43.        else
44.          synapse.permanence -= PERMANENCE_DECREMENT
45.
46.      newSynapseCount = (SAMPLE_SIZE -
47. numActivePotentialSynapses(t-1,
learningSegment))
48.      growSynapses(learningSegment, newSynapseCount)
49.  function punishPredictedColumn(column)
50.    if LEARNING_ENABLED:
51.      for segment in segmentsForColumn(column, matchingSegments(t-1))
52.        for synapse in segment.synapses
53.          if synapse.presynapticCell in activeCells(t-1) then
54.            synapse.permanence -= PREDICTED_DECREMENT

```

図 3-1 3tm algorithm 疑似コード①

出典[6]

```

55. for segment in segments
56.     numActiveConnected = 0
57.     numActivePotential = 0
58.     for synapse in segment.synapses
59.         if synapse.presynapticCell in activeCells(t) then
60.             if synapse.permanence ≥ CONNECTED_PERMANENCE then
61.                 numActiveConnected += 1
62.
63.             if synapse.permanence ≥ 0 then
64.                 numActivePotential += 1
65.
66.         if numActiveConnected ≥ ACTIVATION_THRESHOLD then
67.             activeSegments(t).add(segment)
68.
69.         if numActivePotential ≥ LEARNING_THRESHOLD then
70.             matchingSegments(t).add(segment)
71.
72.     numActivePotentialSynapses(t, segment) = numActivePotential

```

---

```

73. function leastUsedCell(column)
74.     fewestSegments = INT_MAX
75.     for cell in column.cells
76.         fewestSegments = min(fewestSegments, cell.segments.length)
77.
78.     leastUsedCells = []
79.     for cell in column.cells
80.         if cell.segments.length == fewestSegments then
81.             leastUsedCells.add(cell)
82.
83.     return chooseRandom(leastUsedCells)

```

```

84. function bestMatchingSegment(column)
85.     bestMatchingSegment = None
86.     bestScore = -1
87.     for segment in segmentsForColumn(column, matchingSegments(t-1))
88.         if numActivePotentialSynapses(t-1, segment) > bestScore then
89.             bestMatchingSegment = segment
90.             bestScore = numActivePotentialSynapses(t-1, segment)
91.
92.     return bestMatchingSegment

```

```

93. function growSynapses(segment, newSynapseCount)
94.     candidates = copy(winnerCells(t-1))
95.     while candidates.length > 0 and newSynapseCount > 0
96.         presynapticCell = chooseRandom(candidates)
97.         candidates.remove(presynapticCell)
98.
99.         alreadyConnected = false
100.        for synapse in segment.synapses
101.            if synapse.presynapticCell == presynapticCell then
102.                alreadyConnected = true
103.
104.        if alreadyConnected == false then
105.            newSynapse = createNewSynapse(segment, presynapticCell,
106.                INITIAL_PERMANENCE)
107.            newSynapseCount -= 1

```

図 3-1 4 tm algorithm 疑似コード②

出典[6]

### 3.3 Anomaly detection

HTM を用いて異常検知を行う場合に anomaly score、anomaly likelihood を用いる方法がある。図 3-12 は実際にこれらの値を算出して anomaly detection を行った概要図である。anomaly detection は HTM に入力されたデータが異常であるかを検知する工程である。

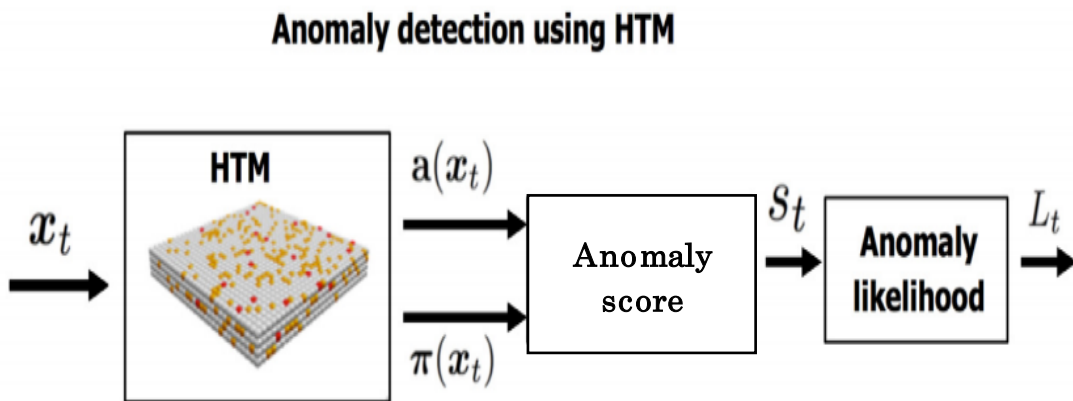


図 3-1 5 anomaly detection 概要図  
出典[1]

HTM ではリアルタイムで発生する連続的なデータストリームを取り扱う。したがって、そうしたデータフレームを以下のように定式化される。(図 3-13 式)

$$\dots, \mathbf{x}_{t-2}, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \dots$$

図 3-1 6 HTM におけるデータフレーム

HTM はこれらのデータを継続的に学習し、その入力の時空間特性をモデル化する。これらの特性は予測に有効であることが示されているが、直接的に異常をモデル化されていない。そのため、異常検知を行うために HTM における 2 種類の内部表現、 $\mathbf{x}_t$ 、 $\boldsymbol{\pi}_t$  を用いる。 $\mathbf{x}_t$  は入力  $\mathbf{x}_t$  に対する疎の表現を表し、 $\boldsymbol{\pi}_t$  は予測ベクトルを表す。この予測ベクトルは現在のシーケンスに関する推論された情報である。予測値の正確さは HTM が現在のデータストリームをどれだけうまくモデリングしているかに依存する。これら 2 つのベクトルから anomaly score ( $s_t$ )、anomaly likelihood ( $L_t$ ) を算出する。初めに、anomaly score は予測入力と実際の入力間の偏差を表し、図 3-14 式で算出する。



$$S_t = \frac{\pi(x_{t-1}) \cdot a(x_t)}{|a(x_t)|}$$

図 3-17 anomaly score

現在の入力完全に予測された場合は 0、予測と完全に違う場合は 1 を示す。また、複数の予測はそれぞれ個別の予測ベクトルで表現される。ベクトルが十分に疎であり、十分な次元であれば適度な数の予測を同時に表現することができ、誤差は指数関数的に小さくなる。anomaly score は現在に入カストリーム予測可能性を瞬間的に測定したものであり、本質的にノイズの多く、実用的アプリケーションでは予測の精度が低下する。特に、anomaly score をそのまま閾値とすると誤検出の割合は高まってしまう。こうした誤検出を防ぐため、anomaly score の分布をモデル化し、anomaly likelihood ( $L_t$ ) を算出する。この指標は現在の状態がどの程度異常であるかを示す。まず、過去  $W$  個の anomaly score の窓を保持し、分布を rolling 正規分布としてモデル化し、標本平均と分散は以前の anomaly score から図 3-15、3-16 のように連続的に更新する。

$$\mu_t = \frac{\sum_{i=0}^{W-1} S_{t-i}}{W}$$

図 3-18 標本平均

$$\sigma_t^2 = \frac{\sum_{i=0}^{W-1} (S_{t-i} - \mu_t)^2}{W-1}$$

図 3-19 分散

次に図 3-17 式で anomaly score の直近の短期平均を計算し、Gaussian 尾部確率（ある変数が平均から標準偏差  $x$  倍以上の値をとる確率）Q-function, [5]、を通じて図 3-18 式で算出する。 $W'$  は短期移動平均の窓であり、 $W' \ll W$ 。これは予測誤差を計算する期間である。 $L_t$  の値の大きさによって異常であるかを判断し、多くの場合閾値が設けられる。

$$\tilde{\mu}_t = \frac{\sum_{i=0}^{W'-1} s_{t-1}}{W'}$$

图 3-2 0 短期平均

$$L_t = 1 - Q\left(\frac{\tilde{\mu}_t - \mu_t}{\sigma_t}\right)$$

图 3-2 1 anomaly likelihood

### 3.4NuPIC

NuPICは numenta 社が開発した HTM 理論に基づく CLA を公式実装したソフトパッケージである。NuPIC には以下の特徴がある。

NuPIC の特徴

- オンライン学習
- ノイズに強い
- 予測、異常検知が可能
- 人間のような知能を持つ
- 汎用性が高い

このパッケージを用いる上で必要となるパラメータは `swarming`[9]で最適化した。下表 1～4 は各パラメータの一覧である。これらの名称は NuPIC が命名したものであり、詳細な理解を深めるためには HTM の文献を参照する必要がある[10]。計算は 8GB の Ram と 4 スレッドの CPU を搭載した 64 ビットの Linux (Ubuntu) マシンで行った。

パラメータ	設定値
<code>clipInput</code>	TRUE
<code>maxval</code>	500
<code>minval</code>	-500
<code>n</code>	396
<code>type</code>	ScalarEncoder
<code>w</code>	21

表 1 `sensorParam`

パラメータ	設定値
alpha	0.075
regionName	SDRClassifierRegion
steps	1
verbosity	0
inferenseType	Temporalanomaly

表 2modelParams

パラメータ	設定値
boostStrength	0
columnCount	2048
globalInhibition	1
inputWidth	0
numActiveColumnsPerInhArea	40
potentialPct	0.8
Seed	1956
SpVerbosity	0
spatialImp	cpp
synPermActiveInc	0.05
synPermConnected	0.1
synPermInactiveDec	0.025

表 3spParams

パラメータ	設定値
acitvationThreshold	15
cellsPerColumn	32
columnCount	2048
golbalDecay	0
intialPerm	0.21
Inputwidth	2048
maxAge	0
maxSegmentsPerCell	128
maxSynapsesPerSegment	32
minThreshold	11
newSynapseCount	20
outputType	Normal
pamLength	4
PermanenceDec	0.1
permanenceInc	0.1
seed	1960
temporalImp	Cpp
verbosity	0

表 4tpParams

## 4.章 実験概要

### 4.1 実験方法

本実験では、刺激によって生じた脳波を異常脳波と定義し、眼球運動、音刺激、光刺激による 3 パターンの実験を行った。アーチファクト対策として、閉眼、安静、暗室の環境下で測定をした。1つ目の実験では、被験者による瞬きを眼球運動とし、4分間の測定時間で適当なタイミングで数回瞬きを促した。2, 3つ目の実験では、3分間の測定時間で最後の1分間に光、音刺激を加えた。脳波測定はフューテックエレクトロニクス株式会社性 BrainPro (ブレインプロ) FM-939 及び PC 用ソフトウェア Analyzer+ (アナライザープラス) FS-AP により行った。BrainPro で検出した脳波を Analyzer+は 1024Hz で原脳波を取得し、高速フーリエ変換 (FFT、窓関数：短形) により、1.0Hz~30.0Hz まで 0.5Hz 毎にパワースペクトル解析する。電極は 10/20 法による FP2 (及び FP1、A1) に装着した。



図 4-1 実験風景

出典

<https://www.futek.co.jp/products/humantech/fm-939/images/fm-939-scene1.jpg>

(2021-07-24 最終ログイン)

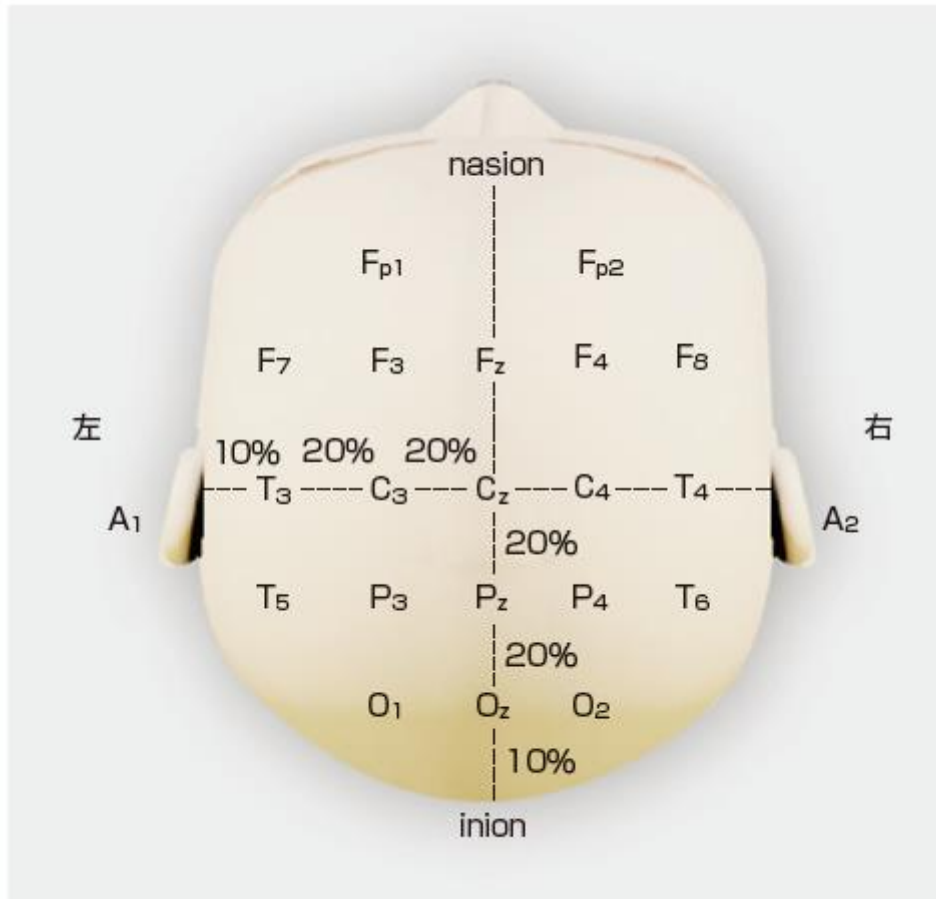


図 4-2 電極位置  
出典[15]

瞬きによる眼球運動によって生じたアーチファクト、音、光刺激によって生じた脳波の変化を異常脳波とし、HTM で学習させることによって異常脳波を検出することを実験目的とした。通常、音、光や痛みなどの刺激に対して脳波は反応を示すため、反応性がない場合は脳に何らかの異常がある可能性がある。

## 5.章 結果と考察

各結果の上部分は実験によって測定した脳波、下部分は HTM によって算出された anomaly likelihood を表す。anomaly likelihood の値が高ければ異常脳波であることを表す。HTM が anomaly likelihood を算出するにあたって序盤の段階は学習が進まないため、数値は安定せず、学習データが増えるにつれて、正確さが増していく。図 5-1 の序盤、EEG の値が安定していない部分は被験者の筋肉の動きによるアーチファクト、中盤以降に大きく値が変化している部分は眼球運動によって生じたアーチファクトである。そのため、序盤に anomaly likelihood の値が大きく変化するのは、学習が進んでいないことに加え、HTM が筋肉の動きによるアーチファクトを異常脳波とも認識しているためである。図 5-1 から瞬きによるアーチファクトが生じた部分に対応して、anomaly likelihood が大きく上昇していることが分かる。これは HTM が異常脳波（眼球運動によって生じたアーチファクト）を検出出来ていることが分かる。しかし、学習が進んだにも関わらず、anomaly likelihood の値は 0 に収束せず、常に何らかの異常入力（異常脳波）があった可能性を表す。これは HTM が様々な要因によって発生したアーチファクトに異常を検出していることが考えられる。anomaly likelihood の値が 0 に近い箇所も存在しており、その部分はアーチファクトによる影響をほとんど受けなかったことが示唆される。このことから HTM は瞬きによるアーチファクトとその他のアーチファクトを識別することが可能であることが分かる。特に今回のような簡易脳波装置を用いる場合にアーチファクトが混入することは容易に想定でき、アーチファクトによる影響を 0 にすることは不可能である。したがって、HTM によって様々なアーチファクトを検出することが出来れば、簡易脳波装置を用いた脳波分析に役立つ。また、近年はスマートウォッチなどの発達により、身近に心拍数などの医療データを測定することが可能となっている。仮に、スマートウォッチなどの手法で脳波を測定できるようになった時にはアーチファクトが多く混入されることが予測できる。そうした場面でも HTM を応用することにより、より精度の高い脳波を検出することが可能となる。異常脳波によって引き起こされるてんかんなどの病気の患者に、日常生活の段階から脳波を測定、異常脳波が発生した際にアラートなどを送ることなどが可能になれば、大いに役立つだろう。



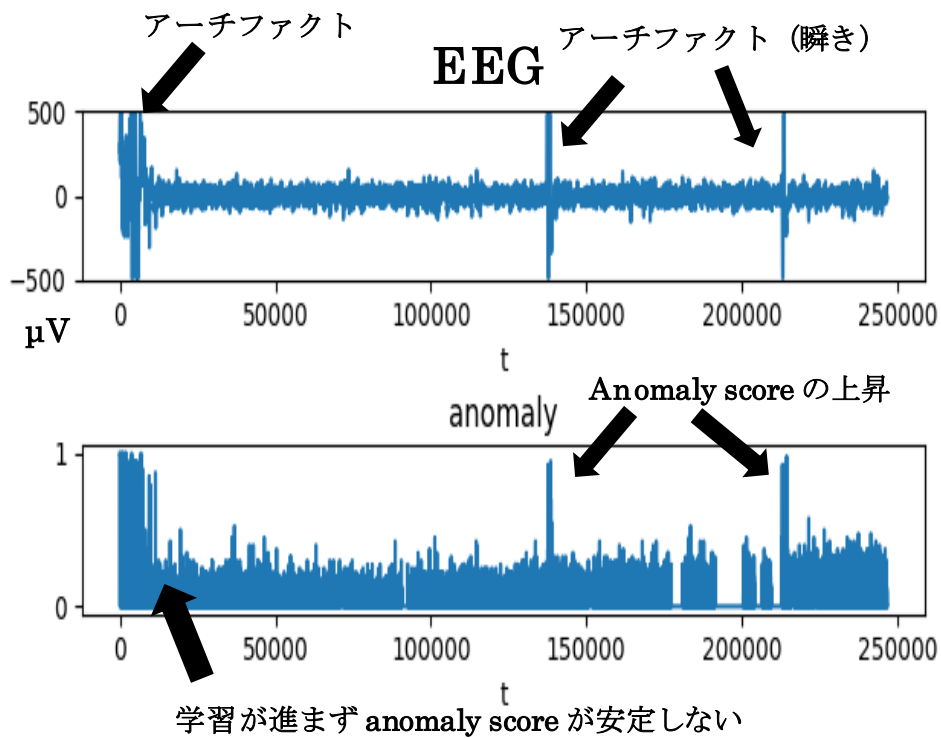


図 5-1 結果①

図 5-2、5-3 は図 5-1 と同様に序盤の段階は学習が進んでいないことに加え、アーチファクトが多く含まれているため、anomaly likelihood は安定していない。しかし、学習がある程度進み、刺激を加えた後にも anomaly likelihood の値は大きく推移しなかった。anomaly likelihood が推移しなかった原因として、そもそも刺激に対して脳波の変化がそこまでなかった、脳波の変化が少なく、アーチファクトによる脳波変化と大差がなかったことが挙げられる。

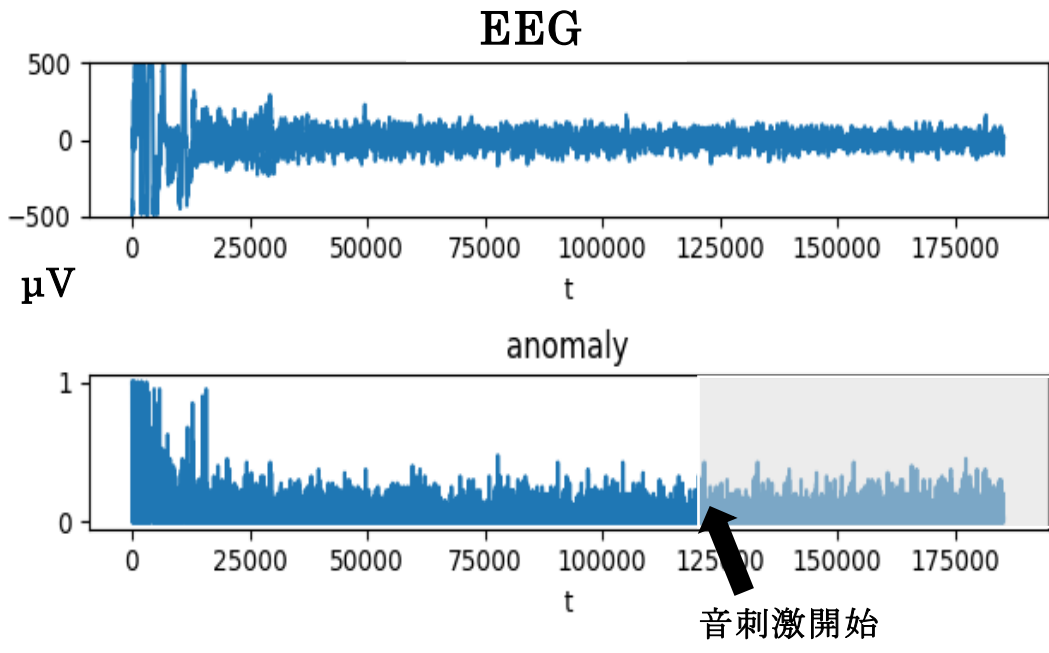


图 5-2 結果②

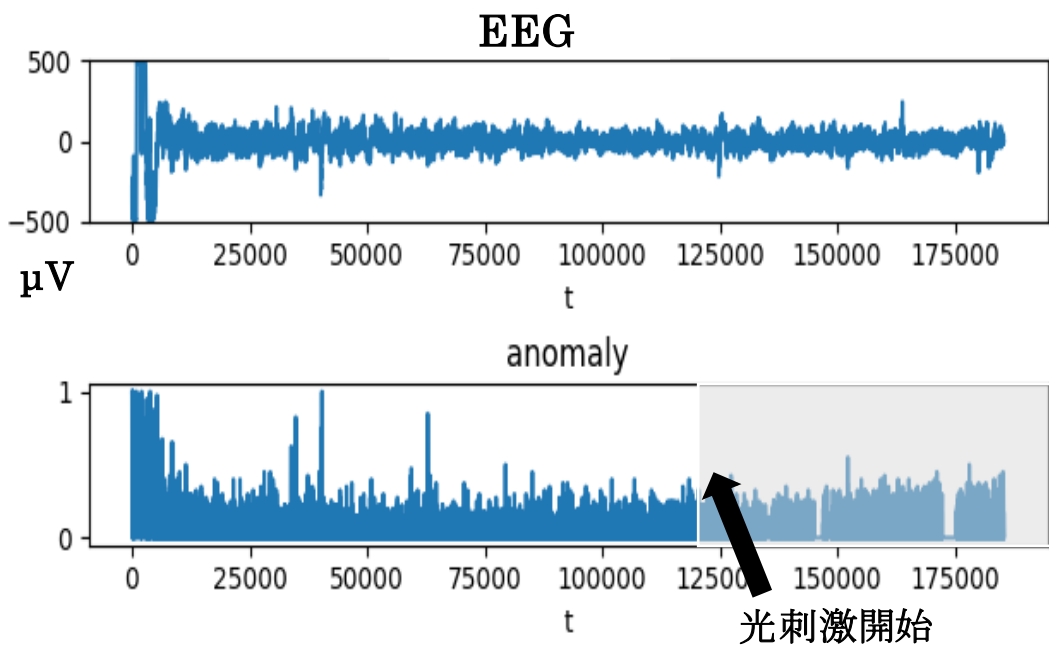


图 5-3 結果③

## 6.章 まとめ

本実験では、HTM によって眼球運動によるアーチファクトを検出することは出来たが、光、音刺激による脳波変化は検出出来なかった。今回の実験では簡易脳波装置を用いたため、アーチファクトによる影響を大きく受けてしまった。したがって、今後の展望としては、アーチファクトによる影響を極力減らすために、より正確な脳波を測定できる機会を用いる、様々な方法でアーチファクトを処理してから、HTM に学習させるなどの方法が考えられる。HTM による研究の数は少なく、実用化という点では、深層学習などの人工知能との間に大きな差がある。しかし、今後数十年先の未来を考えた時に HTM が社会の様々な場面で実装される可能性は高く、目先の実用性だけでなく、将来性という観点を持って研究を進める必要がある。HTM に関する論文のほとんどが海外のものであり、日本のものは数少ない。また、HTM の本質的理解には、生物学、神経学、工学、数学的知識など幅広いものを要する。したがって、今後 HTM の研究する日本の研究者、特に学生の方には本論文を大いに活用して欲しい。

## 7.章 謝辞

本研究を進めていく上でミケレット教授には実験方法、分析方法、実験に対するアドバイスなど様々な分野で熱心な指導をいただきました。

また、ゼミ活動において様々なアドバイスをくださったミケレット研究室の方々にも深く感謝しております。

なお、脳波測定において、**BrainPro** (ブレインプロ) **FM-939** 及び **PC** 用ソフトウェア **Analyzer+** (アナライザープラス) **FS-AP** を貸出させていただいた、フューテックエレクトロニクス株式会社に感謝いたします。

最後に、本研究に携わっていただいた全ての方に感謝の意を示し、謝辞とさせていただきます。

## 8.章 文献

- [1] Ahmad, Subutai, Lavin, Alexander ,Purdy, Scott ,Agha, Zuha. Unsupervised real-time anomaly detection for streaming data. *Nuerocomputing*, 262, p134-137, 2017.8.24
- [2] Alexandre Vivmond. Utilizing the HTM algorithms for weather forecasting and anomaly detection. Master's thesis in software engineering at department of informatics university of Bergen department of computing, mathematics and physics Bergen university college Simula@UiB. 2016.11
- [3]Alshammari Nasser Owaid. Anomaly Detection Using Hierarchical Temporal Memory in Smart Homes. Doctoral thesis, Staffordshire University. 2018.3
- [4] D.E. Padilla, R. Brinkworth, M.D. McDonnell, Performance of a hierarchical temporal memory network in noisy sequence learning, in: *Proceedings of the International Conference on Computational Intelligence and Cybernetics*, IEEE, 2013, pp. 45–51
- [5]G.K. Karagiannidis, A.S. Lioumpas, An improved approximation for the Gaussian Q-function, *IEEE Commun. Lett* 11 (2007) 644–646
- [6] J. Hawkins et al. *Biological and Machine Intelligence (BAMI)*. 2019.12.10  
<https://numenta.com/assets/pdf/biological-and-machine-intelligence/BaMI-Complete.pdf>  
2022年1月24日最終ログイン
- [7] J. Hawkins, S. Ahmad, Why neurons have thousands of synapses, a theory of sequence memory in neocortex, *Front. Neural Circuits*. 10 (2016) 1–13
- [8] J.Mnatzagianian, E. Fokoué, D. Kudithipudi, A Mathematical Formalization of hierarchical temporal memory's spatial pooler, *Front. Robot. AI*. 3 (2017) 81
- [9] M. Taylor. Tutorial on swarming for nupic, 2017
- [10] Numenta. Nupic installation and files (github), 2021
- [11] Ruggero Micheletto, Ahyi Kim. An HTM based cortical algorithm for detection of seismic waves. *Neural and Evolutionary Computing*, 2017.6
- [12] S. Ahmad, J. Hawkins, Properties of sparse distributed representations and their application to Hierarchical Temporal Memory, 2015
- [13] Y. Cui, S. Ahmad, and J. Hawkins. Continuous online sequence learning with an unsupervised neural network model. *Neural Comput.*, 28(11):2474–2504, Nov. 2016. 6.
- [14] 宇城研悟 (2014) アーチファクト対策特集「臨床脳波を行う技術師のために」 p393
- [15] 総監 修 矢崎義雄 内科学第 10 版 2013 朝倉書店
- [16] 飛松省三 (2004) 脳波判読のポイント九州大学大学院医学研究院脳研臨床神経生理  
<https://www.med.kyushu-u.ac.jp/neurophy/point.pdf> 2022年1月24日最終ログイン
- [17] 内藤慎一郎・荻原将文 (2018) 接続セグメントに時間軸を導入した階層的記憶第 34 回 ファジィシステムシンポジウム講演論文集 p253-258
- [18] 発行者 福田正 (2017) AI 白書 2017 人工知能をもたらす技術の革新と社会の変貌

## 付録

付録として本研究で用いたプログラムを添付する。

### Anomaly likelihood 算出

```
#!/usr/bin/python/
# FOR EXECUTING/PUSHING THE NUPIC MODEL
# Before or AFTER SWARMING
# run the model FOR EVER

import time as clk
from nupic.frameworks.opf.model_factory import ModelFactory
from nupic.swarming import permutations_runner
from numpy import *
import matplotlib.pyplot as plt
import pyqtgraph as pg
from pyqtgraph.Qt import QtGui, QtCore
from pyqtgraphptime import time
from model_0 import model_params as mp

#definite word
model_params = mp.MODEL_PARAMS
model = ModelFactory.create(model_params)
model.enableInference({"predictedField": "EEG"})
curve1 = curve2 = curve3 = t = zeros((0))

#data loading
data_file = 'data.csv' # example data
wave = loadtxt('data.csv',delimiter=',',skiprows=1)
def runRealTimeTK(k):
    global curve1, curve2, curve3, t
    # generate real time data and feed them to the HTM
    newEEG = wave[k,1]
# read one by one
    EEG_value= float(newEEG)
    result = model.run({"EEG": EEG_value})
```

```

prediction = result.inferences['multiStepBestPredictions'][1]
anomaly = result.inferences['anomalyScore']
curve1 = append(curve1,EEG_value)
curve2 = append(curve2,prediction)
curve3 = append(curve3,anomaly)
t = append(t,k)

for k in range(x, X): # example data
    runRealTimeTK(k)

#final_graph
fig = plt.figure()

ax1 = fig.add_subplot(3, 1, 1)
ax2 = fig.add_subplot(3, 1, 2)
ax3 = fig.add_subplot(3, 1, 3)

ax1.plot(t, curve1)
ax2.plot(t, curve2)
ax3.plot(t, curve3)

ax1.set_xlabel('t')
ax1.set_title('EGG')

ax2.set_xlabel('t')
ax2.set_title('prediction')

ax3.set_xlabel('t')
ax3.set_title('anomaly')

ax1.set_ylim(-500, 500)
ax2.set_ylim(-500, 500)

fig.tight_layout()
plt.show()

```

## swarming

```
if swarm_label:
    DATA_FILE = 'data.csv' # example data
    SWARM_CONFIG = {
        "includedFields": [
            {
                "fieldName": "EEG",
                "fieldType": "float",
                "maxValue": 500.0,
                "minValue": -500.0
            }
        ],
        "streamDef": {
            "info": "EEG",
            "version": 1,
            "streams": [
                {
                    "info": DATA_FILE,
                    "source": DATA_FILE,
                    "columns": [
                        "*"
                    ]
                }
            ]
        },
        "inferenceType": "TemporalAnomaly",
        "inferenceArgs": {
            "predictionSteps": [
                1
            ],
            "predictedField": "EEG"
        },
        "swarmSize": "medium"
    }
```



```

def swarm_over_data():
    return permutations_runner.runWithConfig(SWARM_CONFIG,
        {'maxWorkers': 4, 'overwrite': True})

if swarm_label:
    dd=raw_input(WARNING+"I am going to SWARM !!!")
    model_params = swarm_over_data()
# <== use this if you swarm for best model
    print(HEADER+"swarm for best model finished !"+ENDC)
else:
    # you have to create __init__.py in folder model_0
    from model_0 import model_params as mp
    model_params = mp.MODEL_PARAMS
# <== use this if you already have a model

```

## パラメータ

```
MODEL_PARAMS = {'aggregationInfo': {'days': 0,
    'fields': [],
    'hours': 0,
    'microseconds': 0,
    'milliseconds': 0,
    'minutes': 0,
    'months': 0,
    'seconds': 0,
    'weeks': 0,
    'years': 0},
    'model': 'HTMPrediction',
    'modelParams': {'anomalyParams': {'u'anomalyCacheRecords': None,
    'u'autoDetectThreshold': None,
    'u'autoDetectWaitRecords': None},
    'clParams': {'alpha': 0.075025000000000001,
    'regionName': 'SDRClassifierRegion',
    'steps': '1',
    'verbosity': 0},
    'inferenceType': 'TemporalAnomaly',
    'sensorParams': {'encoders': {'u'EEG': {'clipInput':
    True,
    'fieldname':
    'EEG',
    'maxval': 500.0,
    'minval': -500.0,
    'n': 396,
    'name': 'EEG',
    'type':
    'ScalarEncoder',
    'w': 21}},
    'sensorAutoReset': None,
    'verbosity': 0},
    'spEnable': True,
    'spParams': {'boostStrength': 0.0,
```

```

        'columnCount' : 2048,
        'globalInhibition' : 1,
        'inputWidth' : 0,
        'numActiveColumnsPerInhArea' : 40,
        'potentialPct' : 0.8,
        'seed' : 1956,
        'spVerbosity' : 0,
        'spatialImp' : 'cpp',
        'synPermActiveInc' : 0.05,
        'synPermConnected' : 0.1,
        'synPermInactiveDec' :
0.025225000000000004},
        'tmEnable' : True,
        'tmParams' : { 'activationThreshold' : 15,
        'cellsPerColumn' : 32,
        'columnCount' : 2048,
        'globalDecay' : 0.0,
        'initialPerm' : 0.21,
        'inputWidth' : 2048,
        'maxAge' : 0,
        'maxSegmentsPerCell' : 128,
        'maxSynapsesPerSegment' : 32,
        'minThreshold' : 11,
        'newSynapseCount' : 20,
        'outputType' : 'normal',
        'pamLength' : 4,
        'permanenceDec' : 0.1,
        'permanenceInc' : 0.1,
        'seed' : 1960,
        'temporalImp' : 'cpp',
        'verbosity' : 0},
        'trainSPNetOnlyIfRequested' : False},
    'predictAheadTime' : None,
    'version' : 1}

```