

# 用語説明

## 1. ニューロン

脳にある神経細胞で、生体の中で情報処理用に特別な分化をとげた細胞である。ニューロンの細胞膜の電位を測定すると、短い時間幅（約 1[ms]）をもったスパイク状の波形が観測される（ニューロンが発火するという）。このスパイクは軸索と呼ばれる長い神経線維を介し、シナプスというニューロン間の結合部分を通じて他のニューロンに伝えられる。

McCulloch と Pitts は、ニューロンが発火しているかしていないかという 2 値の論理素子として捉え、理論的な取り扱いが容易なモデルを提案した。このモデルでは、ニューロンを多入力-1 出力の非線形素子とし、スパイク発生の閾値特性やシナプス入力の空間的加算特性などが導入されている。

## 2. 結合定数

シナプス結合、結合荷重、重みとも呼ばれる。ニューロン間の結合の強さを表わし、情報にどの程度の重みを付加して伝達するか決めるものである。

## 3. ニューラルネットワーク

脳を情報処理システムとみなし、情報が脳の内部でどのように表現、処理されているかをモデル化したものである。まず、ニューロンの出力を決める関数には、通常、飽和特性に対応した非線形関数を用いられる。そして、ネットワーク内における情報拡散の結果、各ニューロンは複数のニューロンから結合定数に従って重みづけされた情報を受け取り、加算する。ここでは、知識は結合定数の集合として表現され、新たな知識の獲得は学習則に基づいた各結合定数の調節によって実現される。このような相互作用の結果、ネットワーク全体で複雑な情報処理が可能となる。これは情報処理技術の 1 つとして、制御理論、数理統計学、パターン認識、最適化など多くの分野で用いられている。

## 4. 階層型ニューラルネットワーク

ニューロンがいくつかの層に分かれており、層の間が入力層から出力層に向かう方向に結合されているものである。入出力層以外の層は、中間層と呼ばれる。中間層が 1 層か 2 層、すなわち、入力層・出力層と合わせて 3 層か 4 層のネットワークがよく用いられる。入力信号は層を経ながら順に処理されていき、出力層

に到達したところで処理が終了する。入力から出力への写像として表現できる問題に適している。

## 5. 相互結合型ニューラルネットワーク

すべてのニューロンが他のすべてのニューロンと結合している。従って、信号は一方向に伝播するのではなく、ニューロン間で相互にやりとりされてネットワーク内を循環する。最終的に一定の状態に収束することもあるが、複数の状態の間を振動したり、カオス的な状態遷移をするような複雑なダイナミクスを示すこともある。多数の情報が相互に制約し合う状況で、最適解を見つける多重制約充足問題に適している。

## 6. 学習

脳の学習は、ニューロン間の結合定数の可塑的な変化によるものである。ニューラルネットワークの学習には、大きく分けて教師あり学習、強化学習、教師なし学習の3つの枠組みがある。

- 1) 教師あり学習: 入力  $x$  に対する教師出力  $y$  の組  $(x,y)$  から、入出力写像  $y=f(x)$  を学習する。小脳は、登上線維入力に応じたシナプスの長期減弱 (LTD) による教師あり学習により、身体や環境の内部モデルを獲得している。
- 2) 強化学習: 出力  $y$  に対して与えられる報酬信号を最大化するように、入出力写像  $y=f(x)$  を探索的に学習する。大脳基底核は、ドーパミン入力に依存したシナプス可塑性による強化学習により、自発的な行動の学習を実現している。
- 3) 教師なし学習: 入力  $x$  の統計的な構造だけから、その要素や成分を抽出した表現  $y=f(x)$  を学習する。大脳皮質は、入出力の相関による Hebb 型のシナプス可塑性により、自己組織化マップ、独立成分分析などの教師なし学習を行っている。

## 7. 連想記憶

普通のコンピュータでは、アドレスを指定することによってデータが取り出される。これに対して、人間は、記憶の一部から関係の深いより多くの事柄を想起することができる。このように、想起したいデータの一部から、完全な記憶パターンを出力することを連想記憶という。

ニューラルネットワークで連想記憶を実現するためには、まずニューロンに与

えられた入力に対し出力を計算する．そして、その結果をフィードバックして入力を修正し、それに対して出力を再計算する．このような処理を何度も繰り返すと、出力値の変化が少なくなり、一定の値をとるようになる．このときのニューロンの状態を想起パターンとみなせば良い．

## 8. 自己組織化マップ

自身を含めた近傍のニューロンへは発火性、その他へは非発火性の結合定数を与えると、最大入力を得たニューロンとその近傍の複数のニューロンが同時に発火ようになる．このような局在発火と競合学習を組み合わせると、入力空間の類似関係を、発火するニューロンの距離関係に変換するトポロジー保存写像が得られる．

T.Kohonen は、このような性質をもつネットワークを競合のダイナミクス計算を行わずに実現する自己組織化マップ (Self-Organizing Map) を提案した．このアルゴリズムでは、まず入力  $u$  に最も類似した重みベクトル (結合定数)  $W$  をもつニューロン  $l$  を求め、これを中心とした近傍領域を決定する．そして、この領域内に含まれるニューロンは同時に発火するものとし、これらの結合定数のみ入力  $x$  に近づくよう修正する．これにより、類似した入力に対して、近い位置にあるニューロンが発火ようになる．つまり、入力の類似関係が、配置されるニューロンの距離関係に変換される．

## 9. Hopfield Model

相互結合型ニューラルネットワークで、結合は対称で自己結合はなく、非同期にニューロンの状態を更新する．更新とともに減少するエネルギー関数があり、これを用いて組合せ最適化問題や連想記憶問題に応用することができる．

連想記憶問題に応用する場合、パターン間の直交性を利用するため、通常、ニューロンの状態を 1 (発火)、または -1 (非発火) とする．結合定数は Hebb 則で与えるが、各記憶パターンは一回学習すれば良い．記憶パターンの想起は、それに近い初期パターンから出発して状態更新を続け、エネルギーが最小になる、すなわちその記憶パターンに収束することで実現される．

## 10. Hebb 則

1949 年に D.O.Hebb は、相互に結合した閉回路を形づくるニューロン群が、1 つの機能単位をなすと考え、これを細胞集成体と名づけた．さらに、シナプスをはさむ 2 つのニューロンの活動が同期したとき、結合は強められるという学習則

(Hebb 則) を主張した.

## 11. カオス

自然界には予測可能な現象が多くあるが、大気の運動のように、物理法則に従っていても、原因と結果の関係が明瞭でない現象もある. このように、比較的簡単な規則に支配されながら、時間とともに不規則的に変化する現象をカオスという. この場合、未来の値は一意的に決定されるが、初期条件に非常に鋭敏に（指数関数的に）依存する性質をもっているため、過去が少しでも異なると結果が大きく変わってくる. 水滴の落下間隔、心拍、株価の変動などもカオスである.

## 12. 間欠性カオス

周期的な振舞いとカオス的な振舞いが、交互に現れる現象をいう. ウィンドウの直前に存在する.

## 13. ストレンジアトラクタ

$t \rightarrow \infty$  で軌道が近づく集合をアトラクタといい、相空間が 2 次元面の場合、軌道を引きつける点や極限軌道と呼ばれる閉曲面が存在することが知られている. カオス的な系では、写像を引き伸ばしたのち、それ自身に重ねる（折りたたむ）ことによって、構造の内部に構造を含んだアトラクタとなる. これをストレンジアトラクタと呼ぶ.

## 14. ロジスティックマップ

カオス的な振舞いを示す代表的な写像で、 $x(n+1)=rx(n)(1-x(n))$  ( $0 \leq x \leq 1$ ,  $0 \leq r \leq 4$ ) で表わされる. ロジスティックマップでは、まず  $r < 1$  のとき、 $x(n)$  は 0 に収束し、 $1 < r < 3$  のとき、他のある 1 点に収束する. このような収束値を固定点という. そして、3 より大きくすると、固定点が不安定化し 2 周期を示すようになり、さらに大きくすると、2 周期が不安定になり 4 周期が現れる. このように  $r$  が増加するにつれ、8 周期、16 周期、32 周期、... といったように周期が増えていき、 $r=3.56$  を超えると無限周期、つまりカオス的な振舞いを見ることができる.

## 15. 分岐

力学系において、あるパラメータが一定の値を超えると解の数が変わり、解が定性的に変化する現象を分岐という. また、このパラメータを分岐パラメータという. 特に、ロジスティックマップで観察されるような、分岐パラメータの増加

とともに解の周期が 2, 4, 8, 16, ... と倍に増加していく現象を, 周期倍化分岐という.

## 16. ウィンドウ

分岐図には, カオス領域の中に周期的になる領域が含まれている. このような領域をウィンドウという. ウィンドウで周期的な解となった後は, 再び周期倍化分岐を起こし, カオスとなる. ロジスティックマップにおいては,  $3.56 < r < 4$  のうち約 10% をウィンドウが占めている.

