深層学習の源流

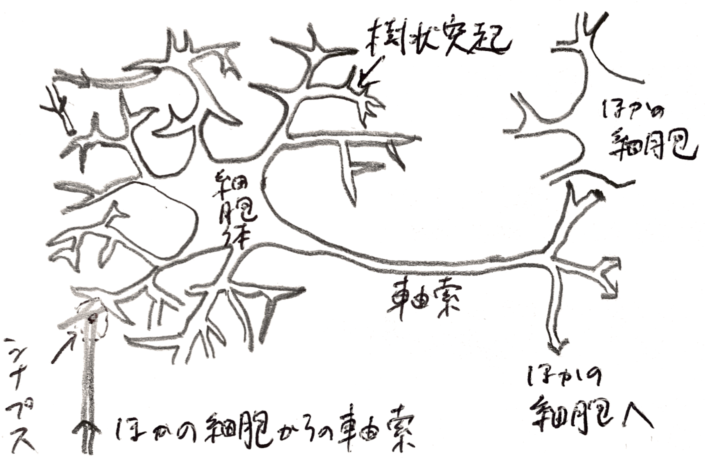
脳からパーセプトロンへ

Tatsuro Senga

Keio

神経細胞とその結合

甘利俊一, 1978, 神経回路網の数理



数理脳科学の視点

目的

深層学習の歴史と理論的進展を、**数理脳科学**の観点から概観する。

数理脳科学とは

生物学的現実性のみに固執せず、単純化された数理モデルを通じ、脳の情報処理における基本原理の理解を目指す学問分野である。

- 脳は進化の過程で形成された複雑なシステムであり、非効率な部分を含む可能性がある。
- そのため、本質的な原理を抽出し、知能や意識の起源を解明することが重要となる。
- 経済学のアプローチと似ているところがある。

AI の黎明期：二つのアプローチ

AI の歴史は、大きく二つの異なる潮流から始まった。

1. 論理ベースの AI

- 1956 年のダートマス会議が起源 (John McCarthy, Marvin Minsky, Claude Shannon, etc.).
- コンピュータを、論理推論を行うチューリングマシンと見なすアプローチ。
- Alan Turing, 1950, Computing Machinery and Intelligence.

2. 脳神経科学に触発された AI

- 心理学者 **F. Rosenblatt** が提唱。
- 知能は神経回路の学習プロセスから生まれるという思想。
- **パーセプトロン**の開発へ繋がる。
- F. Rosenblatt, 1961, Principles of Neurodynamics.

McCulloch-Pitts モデルからパーセプトロンへ

McCulloch-Pitts モデル (1943)

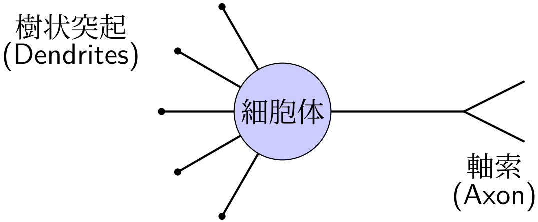
- 脳神経科学に基づく、初のニューロン数学モデル。
- **仕組み**: 複数の二値入力 (0 or 1) を受け取り、その合計が固定のしきい値を超えた場合に 1 を出力する。
- **限界**: 重みやしきい値は手動で設定する必要があり、学習する能力を持たない。

パーセプトロン (1961)

Rosenblatt が McCulloch-Pitts モデルに学習則を導入したモデル。

- **仕組み**: 複数の入力に**重み**を付け、その合計値が**しきい値**を超えると発火する。
- **学習**: 出力が誤っていた場合、その誤りに基づいて重みを自動で調整する。これにより、データから自動で分類境界を学習できるようになった。
- **能力**: 線形分離可能なパターンであれば、有限回の学習で必ず分類できる (収束定理).

神経細胞とその結合



生物学的ニューロン：

- 複数の信号を受信
- 情報を統合
- 閾値を超えると発火

数学的モデル：

- 重み付き入力を合計
- バイアス項を追加
- 活性化関数を適用

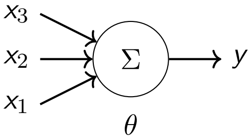
出典：甘利俊一, 1978, 神経回路網の数理

特徴：

- ニューロンの最初の数学的モデル
- バイナリ入力（0 または 1）
- 固定閾値 θ
- バイナリ出力
- 学習能力なし！

数学的形式：

$$y = \begin{cases} 1 & \text{if } \sum_i x_i \geq \theta \\ 0 & \text{otherwise} \end{cases}$$



Rosenblatt のパーセプトロン

革命的特徴：

- 調整可能な重み！
- 学習アルゴリズム
- 収束定理
- 自動分類

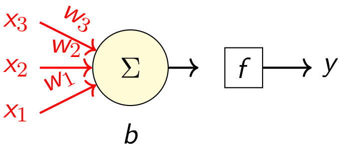
学習ルール：

- 出力が正しい場合：何もしない
- 出力=0、正解=1：重みを増加
- 出力=1、正解=0：重みを減少

数学的形式：

$$u = \sum_{i=1}^{n} w_i x_i + b$$

$$y = f(u)$$



パーセプトロンの限界：XOR 問題

ミンスキーとパパートによる批判 (1969)

Perceptrons: An Introduction to Computational Geometry (1969) で、単層パーセプトロンの数学的な限界を示した。

- **XOR 問題**のような、線形分離不可能な問題は原理的に解けないことを示した。
- この批判は、ニューラルネットワーク研究の資金提供を停滞させ、「冬の時代」を招く一因となった。
- 他にも、パーセプトロンの学習則には、計算量が多く、実用的でないという問題があった。

実用例：Gitでのファイル差分検出

2つのバージョンの違いを見つける

XORで差分検出

ファイルA: 1010
ファイルB: 1100

XOR結果: 0110

結果の意味：

- 0 = 同じビット
- 1 = 異なるビット

→ 2番目と3番目のビットが変更された

XORの真理値表

旧	新	XOR	意味
0	0	0	変更なし
0	1	1	変更あり
1	0	1	変更あり
1	1	0	変更なし

git diffの基本原理
（実際はもっと複雑ですが、
差分検出の基本はXOR）

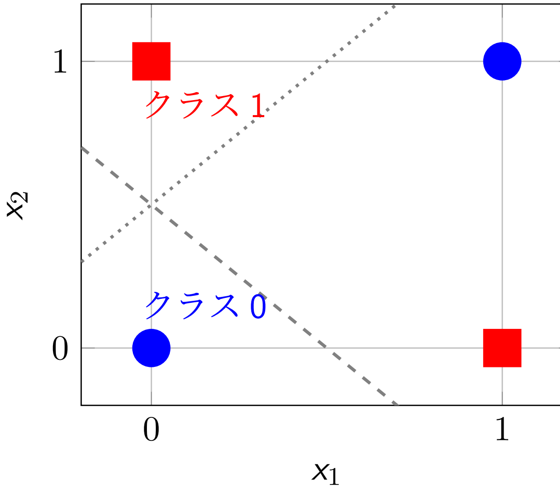
問題：このパターンをニューラルネットワークで学習できるか？

XOR 真理値表：

x_1	x_2	出力
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

問題：
単一の線でクラスを分離できない！

XOR は線形分離不可能



数学的な矛盾 (The Mathematical Contradiction)

パーセプトロンの出力は、$w_1x_1 + w_2x_2 + b$ の符号による。

出力 1 を「> 0」、出力 0 を「≤ 0」と仮定すると、4 つの条件が生まれます。

1. **入力 (0, 0) → 0:** $w_1(0) + w_2(0) + b \leq 0 \implies \boldsymbol{b \leq 0}$
2. **入力 (0, 1) → 1:** $w_1(0) + w_2(1) + b > 0 \implies \boldsymbol{w_2 + b > 0}$
3. **入力 (1, 0) → 1:** $w_1(1) + w_2(0) + b > 0 \implies \boldsymbol{w_1 + b > 0}$
4. **入力 (1, 1) → 0:** $w_1(1) + w_2(1) + b \leq 0 \implies \boldsymbol{w_1 + w_2 + b \leq 0}$

矛盾 (Contradiction)

- 条件 2 から $w_2 > -b$。
- 条件 3 から $w_1 > -b$。
- この 2 つを足すと: $\boldsymbol{w_1 + w_2 > -2b}$

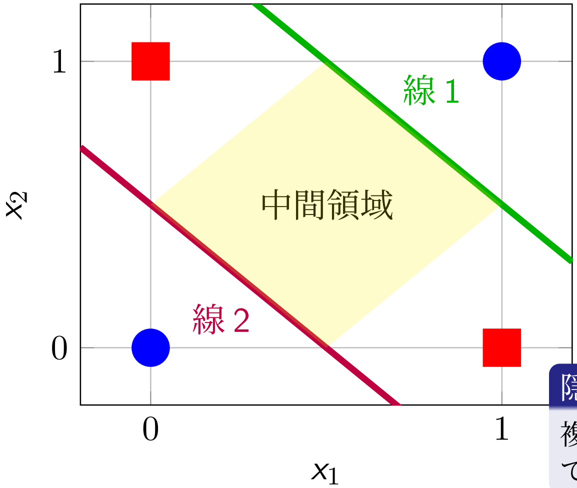
しかし、条件 4 は $\boldsymbol{w_1 + w_2 \leq -b}$ を要求。

これらの条件を同時に満たす重み (w_1, w_2) とバイアス (b) は存在しない。

複数の直線を組み合わせる

2 層ネットワークの戦略：

2 つの直線で領域を分割



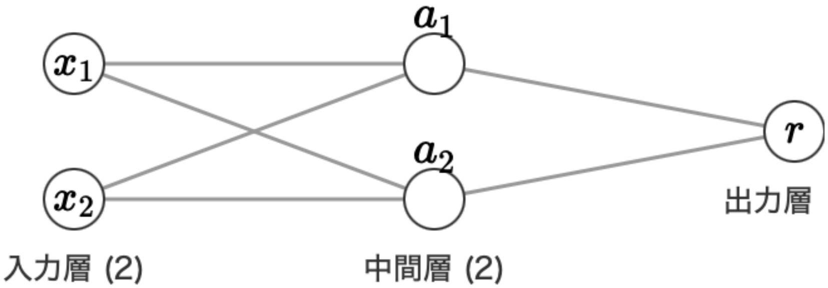
隠れ層の役割：

1. **ニューロン 1：**
   $h_1 = f(x_1 - x_2 + 0.5)$
   左上を検出
2. **ニューロン 2：**
   $h_2 = f(x_1 - x_2 - 0.5)$
   右下を検出
3. **出力層：**
   $y = f(h_1 - h_2)$
   中間領域を抽出！

隠れ層

複数の線形境界を組み合わせて非線形な決定境界を作成

ネットワークの構造と計算ルール



3 層構造

- **入力層 (x_1, x_2)**: データを受け取る
- **中間層 (a_1, a_2)**: 入力を新表現に変換
- **出力層 (r)**: 最終的な答えを出力

XOR 問題の真偽表

XOR 問題の真偽表

入力層		中間層		出力層
x_1	x_2	a_1	a_2	r
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

計算式

- $a_1 = x_1 \times 1 + x_2 \times 1 - 0.5$
- $a_2 = x_1 \times -1 + x_2 \times -1 + 1.5$
- $r = a_1 \times 1 + a_2 \times 1 - 1.5$

計算結果

- 1: 入力 (x1, x2)=(0, 0) → (a1, a2)=(0, 1) → 出力 r: 0
- 2: 入力 (x1, x2)=(0, 1) → (a1, a2)=(1, 1) → 出力 r: 1
- 3: 入力 (x1, x2)=(1, 0) → (a1, a2)=(1, 1) → 出力 r: 1
- 4: 入力 (x1, x2)=(1, 1) → (a1, a2)=(1, 0) → 出力 r: 0

具体的な計算の流れ (1/2)

ケース 1：入力 (0, 0)

- a_1: $(0\times1)+(0\times1)-0.5=-0.5\to\mathbf{0}$
- a_2: $(0\times-1)+(0\times-1)+1.5=1.5\to\mathbf{1}$
- r: $(0\times1)+(1\times1)-1.5=-0.5\to\mathbf{0}$

最終出力: 0

ケース 2：入力 (0, 1)

- a_1: $(0\times1)+(1\times1)-0.5=0.5\to\mathbf{1}$
- a_2: $(0\times-1)+(1\times-1)+1.5=0.5\to\mathbf{1}$
- r: $(1\times1)+(1\times1)-1.5=0.5\to\mathbf{1}$

最終出力: 1

具体的な計算の流れ (2/2) と結論

ケース 3：入力 (1, 0)

- a_1: $(1 \times 1) + (0 \times 1) - 0.5 = 0.5 \rightarrow \mathbf{1}$
- a_2: $(1 \times -1) + (0 \times -1) + 1.5 = 0.5 \rightarrow \mathbf{1}$
- r: $(1 \times 1) + (1 \times 1) - 1.5 = 0.5 \rightarrow \mathbf{1}$

最終出力: 1

ケース 4：入力 (1, 1)

- a_1: $(1 \times 1) + (1 \times 1) - 0.5 = 1.5 \rightarrow \mathbf{1}$
- a_2: $(1 \times -1) + (1 \times -1) + 1.5 = -0.5 \rightarrow \mathbf{0}$
- r: $(1 \times 1) + (0 \times 1) - 1.5 = -0.5 \rightarrow \mathbf{0}$

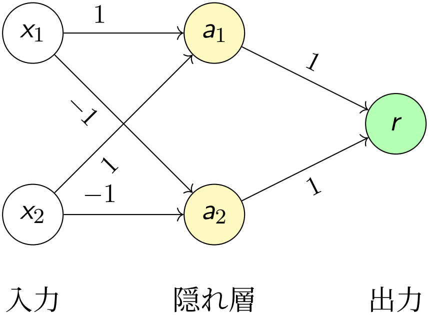
最終出力: 0

中間層が役立つ

もとの入力 (x_1, x_2) は線形分離不可能だったが、中間層がそれらを新しい表現 (a_1, a_2) に変換した。この新しい特徴空間では、問題が線形分離可能になっている。

3 層ネットワークが XOR を解決

ネットワーク構造：



計算：

- $a_1 = x_1 + x_2 - 0.5$
- $a_2 = -x_1 - x_2 + 1.5$
- $r = a_1 + a_2 - 1.5$

検証：

x_1	x_2	a_1	a_2	r
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

完璧な XOR 計算！

中間層学習の基礎：確率的勾配降下法

歴史的発展：今日の最適化手法の基礎は 1950-60 年代に

- **Robbins-Monro (1951)**: 確率的近似法（0）
- **Kiefer-Wolfwitz (1952)**: 確率的近似法（最小値）

これらの**確率的近似法**は、ノイズを含むデータから関数の最小値を逐次的に見つける手法の基礎を築いた。

中間層学習の基礎：アナログパーセプトロンへの応用

多層パーセプトロンの中間層学習則として、オンライン確率的勾配降下法を提案 (ノンパラメトリックな学習法)

- **Amari, 1966, 1967**
- **Tsypkin, 1966**

$$y = f(x, \theta) + \varepsilon$$

$$\text{Loss: } \ell(x, \theta) = \frac{1}{2}\{y - f(x, \theta)\}^2$$

$$\text{Learning: } \Delta\theta = -\eta \frac{\partial \ell(x, \theta)}{\partial \theta}$$

アナログパーセプトロンは連続変数なので微分可能

収束の証明、精度、速度についても証明

甘利俊一 (1968), 学習識別の理論

情報科学講座 A·2·5

情報理論 II

—情報の幾何学的理論—

北川敏男 編

甘利俊一 (1968), 学習識別の理論

確率的降下法（確率的近似法の応用）

- 損失関数 $\ell(x,\theta)$ を最小化したい。しかし $\ell(x,\theta)$ は未知。
- $\theta' = \theta + \Delta\theta$ とすると,$\ell(x,\theta)$ が「平均」として減少するように $\Delta\theta$ を選ぶ。
- これは、坂の途中に立たされた「よっぱらい」が酔歩を行なう状況に似てい る。彼は、あるときは山を登る方向によろめき、また次には山を下る方向によ ろける。登る確率よりは下る確率のほうが大きいため、彼は平均としては山をずり落ちて行き、ついには谷底へ落ち込む。そして谷底の近傍で、よろめき (微小振動) を続けるであろう。（甘利, 1968）

深層学習の夜明け：誤差逆伝播法

「冬の時代」は、1980 年代に**誤差逆伝播法 (Backpropagation)** が再発見・普及したことで終わりを告げる。

誤差逆伝播法 (PDP) とは？

多層ニューラルネットワークを効率的に訓練するアルゴリズム。

- **仕組み**: 損失関数に対する勾配を、出力層から入力層へ効率的に計算する。
- **重要性**: これにより中間層の学習が可能となり、深層学習の発展の礎となった。

主な貢献者

- **Rumelhart, Hinton, Williams (1986)**: PDP(Parallel Distributed Processing) として広く普及させた。
- **Amari (1966), Tsypkin (1966), Werbos (1974)**: それ以前に同様のアイデアを独立して発表。

深層学習ブーム (2012年〜)

現在のブームは、2012年の **ImageNet コンペティション**での歴史的な勝利がきっかけとなった。

ブームの要因

- **計算能力の向上**: GPU の活用。
- **ビッグデータ**: 大規模な訓練データセットの利用。
- **技術革新**: より深いアーキテクチャや正則化技術の開発。

主な応用分野

- 画像認識
- 音声認識
- 自然言語処理

深層学習の課題

データから「実験式」を見つけるのは得意だが、その背後にある根本原理を理解する能力はまだない。

最近の理論的展望：深層学習の謎

深層学習の成功は、従来の統計的学習理論では説明が難しい現象を伴う。

謎 1：過学習の問題

- **従来の理論**: パラメータ数がデータ数を上回ると、過学習が起きるとされた（U 字カーブ現象）。
- **深層学習の現実**: パラメータ数がデータ数をはるかに超えても、高い汎化性能を示す。

謎 2：局所最適解の問題

- **従来の懸念**: 巨大なパラメータ空間には多数の局所最適解が存在し、学習が停滞すると考えられていた。
- **深層学習の現実**: 実際には、局所最適解が問題になることは少なく、多くの場合で性能の良い解に到達する。