



※警告：本資料に解答そのものは記載
されていません。自力で解くための
「思考の武器」を提供します。

単純パーセプトロン・プログラム解析

設問3 [ア]～[エ] 攻略のための完全ヒント集

Mission: 関数 simple_perceptron を完成させよ

```
1 function simple_perceptron(X, Y)
2     for( out を 0 から NC - 1 まで 1 ずつ増やす )
3         ytemp ← [ア]
4
5         ytemp ← ytemp + [イ] × [ウ]
6     endfor
7     if( ytemp が [エ] )
8         Y[out] ← 1
9     else
10        Y[out] ← 0
11    endif
12 endfor
13 endfunction
```

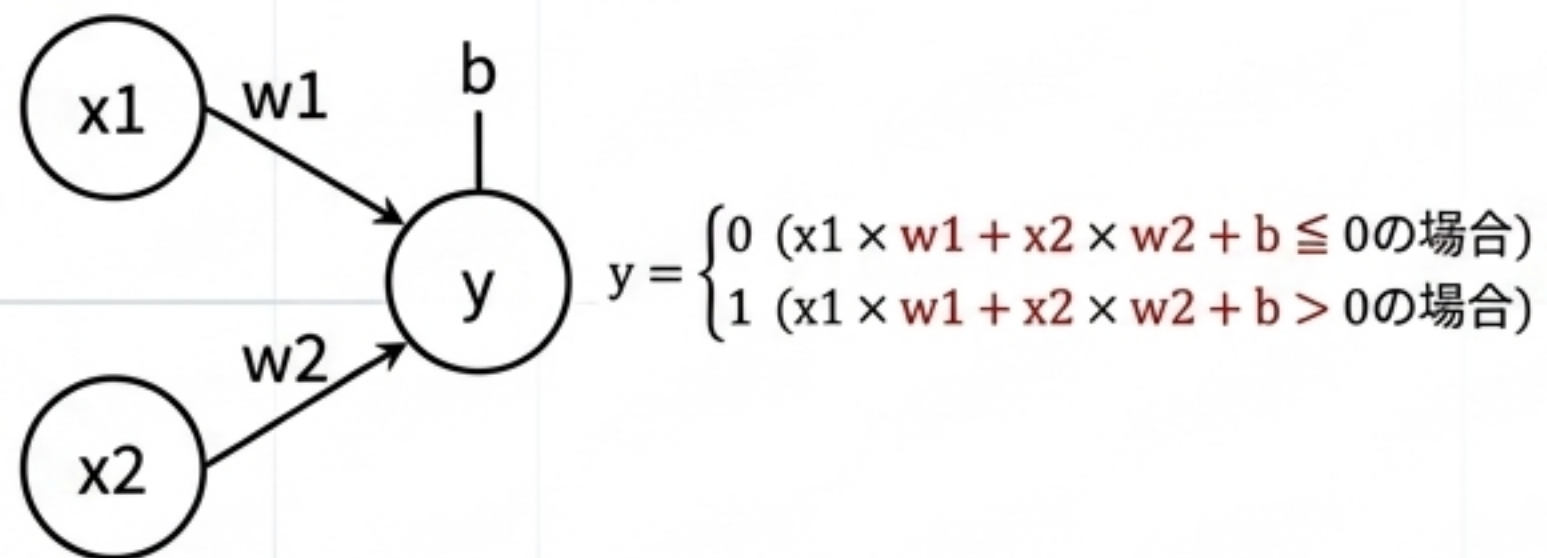
ターゲットは4箇所。論理演算の入力値の全組合せ X から、出力値 Y を導き出すアルゴリズムを復元する。



コードを漫然と眺めるのはここで終わりだ。背景にある「数式」を「プログラム」へ翻訳するプロセスを開始する。

攻略の絶対法則：「数式」と「データ」のマッピング

数学の世界（ノード演算式）



翻訳
(Mapping)

プログラムの世界（変数・配列）

X	論理演算の入力値の全ての組合せを表す2次元配列
Y	論理演算の出力値（演算結果）を格納する1次元配列
WY	ノードyのアーキがもつ重みの値を表す1次元配列
BY	ノードyのバイアスの値（b）を表す変数

- この問題を解く鍵はただ一つ。
- 「ノードの演算式」の各要素が、「どの変数・配列」に格納されているかを完全に結びつけること。

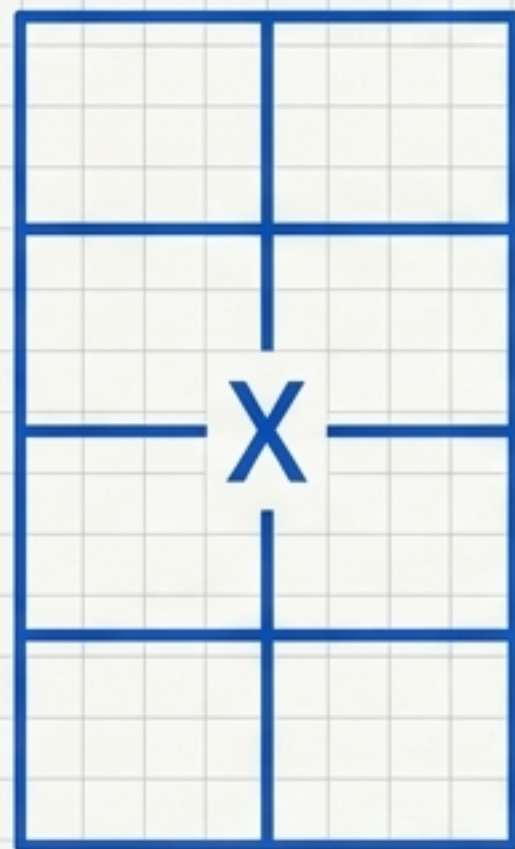
証拠品①：数式モデルの分解

$$y = \begin{cases} 0 & (x_1 \times w_1 + x_2 \times w_2 + b \leq 0 \text{ の場合}) \\ 1 & (x_1 \times w_1 + x_2 \times w_2 + b > 0 \text{ の場合}) \end{cases}$$

入力 × 重みの合計 バイアス値の加算 発火の条件判定

出力 y が決まるメカニズムは3ステップ。①積和を計算し、②バイアスを足し、③条件で判定する。

証拠品②：使えるパーツ（データ構造）



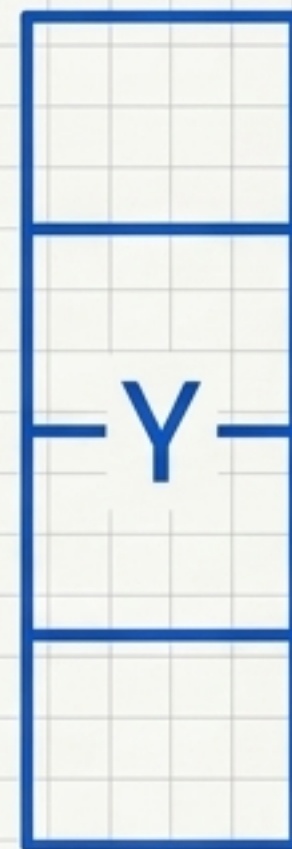
入力値の組合せ
($NC \times NI$)



アークの重み



バイアス値



出力値

ヒント：パーツの「次元数」に注目せよ。単一の値か、1次元か、2次元か？

二重ループの正体：マトリックスの歩き方

for (out ... NC-1) 行のループ：全 4 パターンのテストケースを切り替え



for (in ... NI-1) 列のループ：x1, x2 とノードを順番に処理



外側の out は「どの組合せか」、内側の in は「どの入力ノードか」を指し示している。

Target [ア]：計算の「初期値」

Code World

ytemp ← [ア]

ytemp は最終的なノードの計算結果を保持する変数。この直後の内側ループ (in) で、「入力 × 重み」が次々と ytemp に足し込まれていく。

Math World

$\$ + b\$$



思考のヒント：足し算のループを始める「前」に、あらかじめ ytemp にセットしておくべき基礎値は何か？ 証拠品②の中で、単一の変数として定義されているものは？

Target [イ][ウ] (1/2) : 何を掛けるのか？

Code World

```
ytemp <- ytemp + [イ] * [ウ]
```

Math World

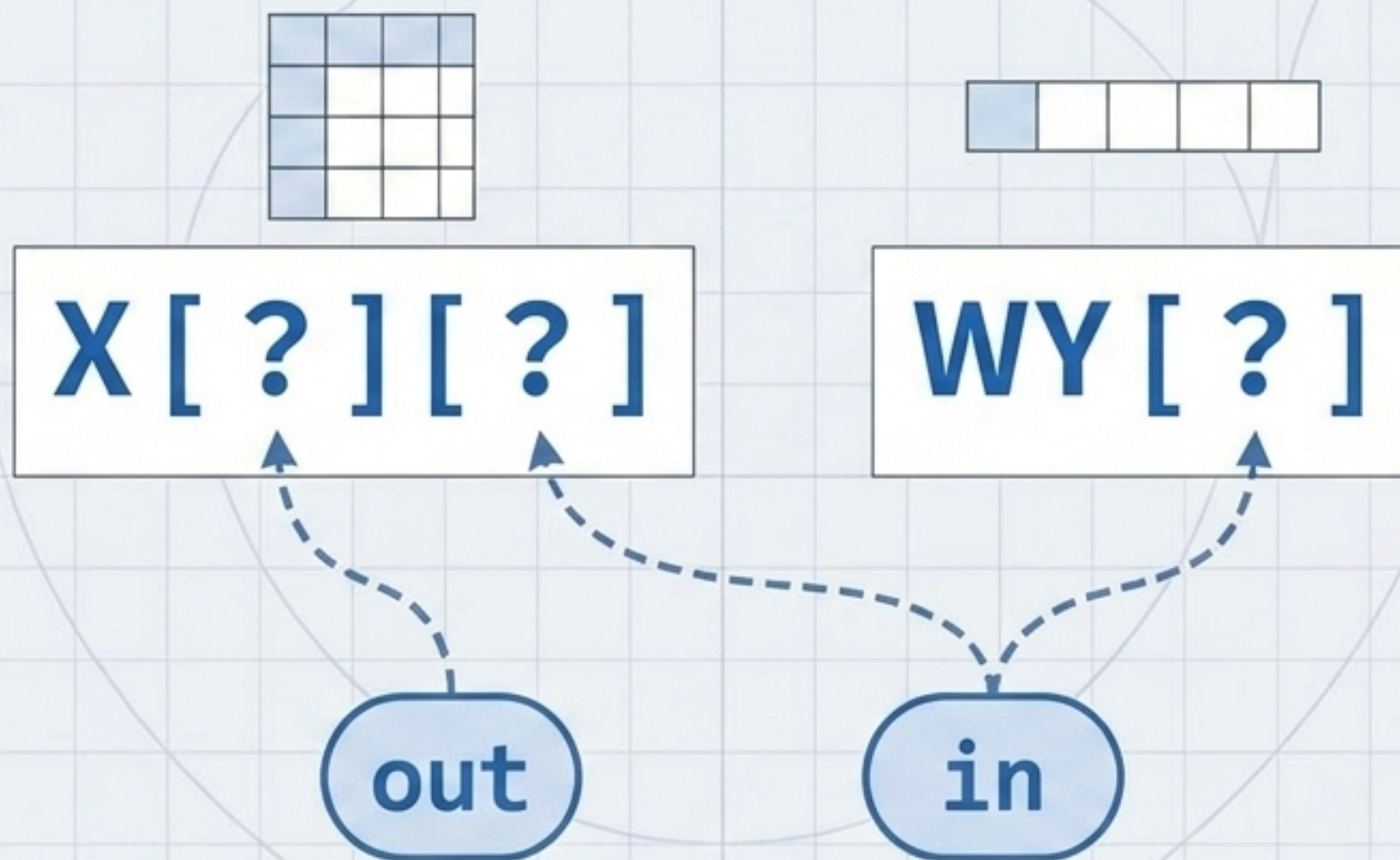
$$x_i \times w_i$$

内側のループは、まさに積和演算の中心部。
片方には「入力値を示す配列」、もう片方には
「重みを示す配列」が入る（順序は問わない）。

 **思考のヒント**：証拠品②のリストから「入力値の組合せ」と「アークの重み」を格納している配列名をそれぞれ見つけ出そう。

Target [イ][ウ] (2/2) : 正しい「添字」の指定

入力値(2次元):
「何番目のテスト
ケース(out)」の
「どのノード(in)」
か、2つの座標が
必要。



重み(1次元):
重みはノードごとに
決まっている。
組合せ番号(out)
は関係なく、「ど
のノード(in)」か
だけを指定する。

💡 **思考のヒント** : ループ変数 out と in を、それぞれの配列の [] の中に正しく
セットしよう。

Target [工]：発火の条件式

Code World

```
if ( ytemp が [工] )  
    Y[out] <- 1
```

Math World

$1 \text{ (} x1 \times w1 + x2 \times w2 + > 0 \text{ の場合)}$



全ての積和計算とバイアス加算が終わり、ytemp の値が確定した。
最後は出力 Y[out] を 1 にする条件の判定だ。



思考のヒント：図3の数式を確認せよ。計算結果が「どういう状態」のときに、出力は1になっているか？その数学記号をそのままプログラムの条件式として書き込もう。

ミッション最終確認：全ピースの結合

☒ `ytemp` の初期値には、単一変数の「**バイアス**」を指定したか？ (Target ア)

☒ 内側のループで「入力の2次元配列」と「重みの1次元配列」を掛け合わせたか？

☒ その際、配列の添字としてループ変数の `out` と `in` を正確に割り当てたか？
(Target イ, ウ)

☒ 最後の `if` 文は、数式モデルの「出力が1になる条件」と完全に一致しているか？
(Target エ)

すべての証拠は揃った。解答を組み立てよ！