パーセプトロン

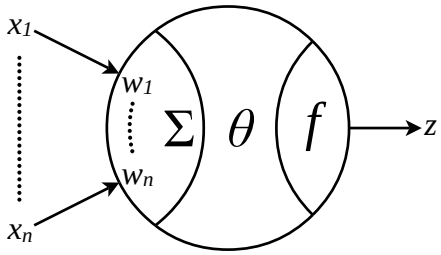
（ニューラルネットワーク入門）

cf. https://wwws.kobe-c.ac.jp/~miura/stock/#ai

ニューロセルの基本構造(1)

マカロック=ピッツ モデル

x_1〜x_n ：入力（アナログ値もOK）
w_1〜w_n ：重み
θ ：閾値
f ：活性化関数（ステップ関数）



$$f(x) = \begin{cases} 0, & (x < 0) \\ 1, & (x \geqq 0) \end{cases}$$

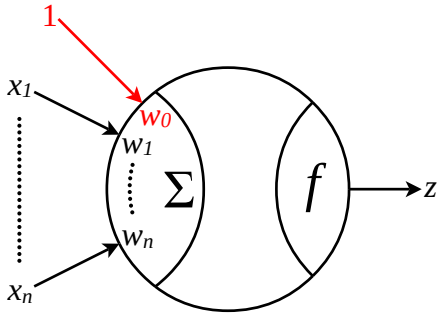
$$u = \sum_{x=1}^{n} w_i x_i - \theta \quad (= w_1 x_1 + \cdots + w_n x_n - \theta)$$

$z = f(u)$ （u：活性値，z：出力（0 or 1））

ニューロセルの基本構造(2)

現代的なモデル

x_1〜x_n ：入力（アナログ値もOK）
w_1〜w_n ：重み
$w_0\ (=-\theta)$ ：バイアス（$x_0 = 1$と考える）
f ：活性化関数
（シグモイド関数，ReLU等）

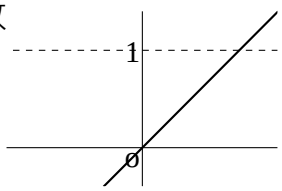


$$u = \sum_{x=0}^{n} w_i x_i \quad (= w_0 + w_1 x_1 + \cdots + w_n x_n)$$

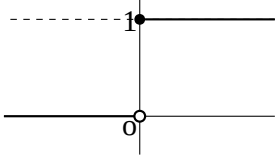
$z = f(u)$ （u：活性値，z：出力（アナログ値もOK））

活性化関数

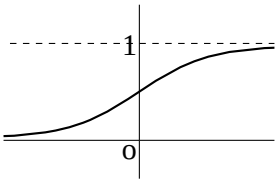
- 恒等関数



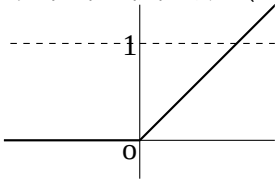
- ステップ関数



- シグモイド関数



- 正規化線形関数 (ReRU)



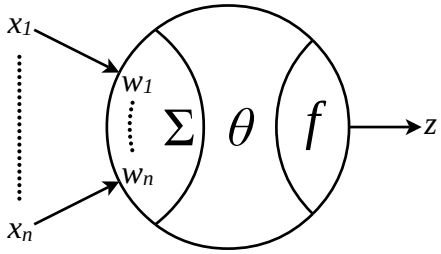
活性化関数

- 恒等関数 → 線形近似式
  - 多層化が無意味
- ステップ関数 → 線形判別式
- シグモイド関数：可微分
- 正規化線形関数 (ReRU)：片側可微分
- 他（さまざまに工夫されている）

ニューロセルの例

（マカロック=ピッツ モデル で考える）

x_1〜x_n ：入力
w_1〜w_n ：重み
θ ：閾値
f ：活性化関数（ステップ関数）



$$u = \sum_{x=1}^{n} w_i x_i - \theta \quad (= w_1 x_1 + \cdots + w_n x_n - \theta)$$

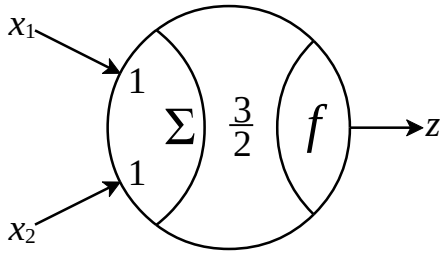
$$z = f(u)$$

活性値（x_iの一次式）
fと組み合わせることで，
線形判別式として機能する

ニューロセルの例 (2入力の場合(1))

$w_1 = w_2 = 1$
$\theta = \frac{3}{2}$

の場合,



$x_1 + x_2 - \frac{3}{2} < 0$ のとき $z = 0$

$x_1 + x_2 - \frac{3}{2} \geqq 0$ のとき $z = 1$

ニューロセルの例 (2入力の場合(1))

$w_1 = w_2 = 1$
$\theta = \frac{3}{2}$

の場合,

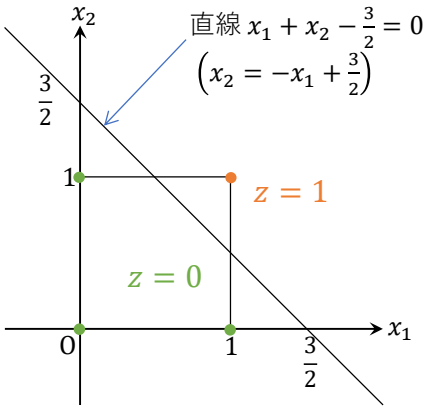
$x_1 + x_2 - \frac{3}{2} < 0$ のとき $z = 0$

$x_1 + x_2 - \frac{3}{2} \geqq 0$ のとき $z = 1$

x_1	x_2	z
0	0	0
0	1	0
1	0	0
1	1	1

z ： x_1 かつ x_2

ニューロセルの例 (2入力の場合(1))



x_1	x_2	z
0	0	0
0	1	0
1	0	0
1	1	1

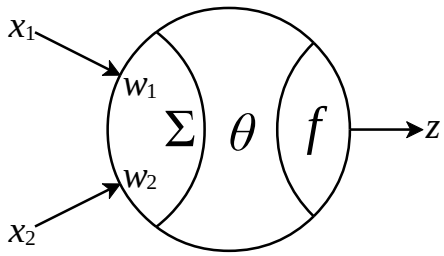
z ： x_1 かつ x_2

重み(w_i)と閾値(θ)によって境界の直線が定まる

ニューロセルの例 (2入力の場合(2))

x_1	x_2	z
0	0	1
0	1	1
1	0	0
1	1	1

z ： x_1 ならば x_2



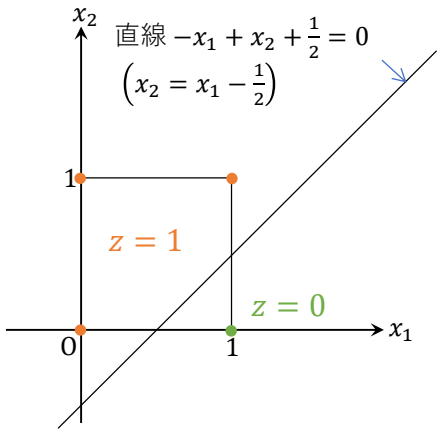
適切な w_1, w_2, θ の値を考えてみよう

ニューロセルの例 (2入力の場合(2))

x_1	x_2	z
0	0	1
0	1	1
1	0	0
1	1	1

z ： x_1 ならば x_2

$w_1 = -1$
$w_2 = 1$
$\theta = -\frac{1}{2}$



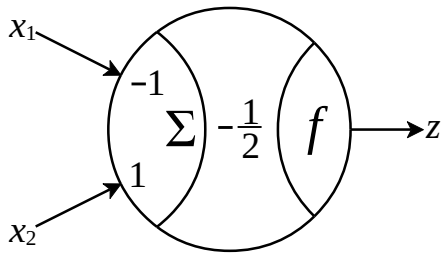
適切な w_1， w_2， θ の値を考えてみよう

ニューロセルの例 (2入力の場合(2))

x_1	x_2	z
0	0	1
0	1	1
1	0	0
1	1	1

z ： x_1 ならば x_2

$w_1 = -1$
$w_2 = 1$
$\theta = -\frac{1}{2}$



適切な w_1， w_2， θ の値を考えてみよう

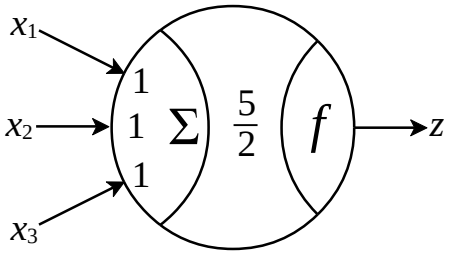
ニューロセルの例 (3入力の場合(1))

$w_1 = w_2 = w_3 = 1$
$\theta = \frac{5}{2}$

の場合,

$x_1 + x_2 + x_3 - \frac{5}{2} < 0 \rightarrow z = 0$

$x_1 + x_2 + x_3 - \frac{5}{2} \geqq 0 \rightarrow z = 1$



ニューロセルの例 (3入力の場合(1))

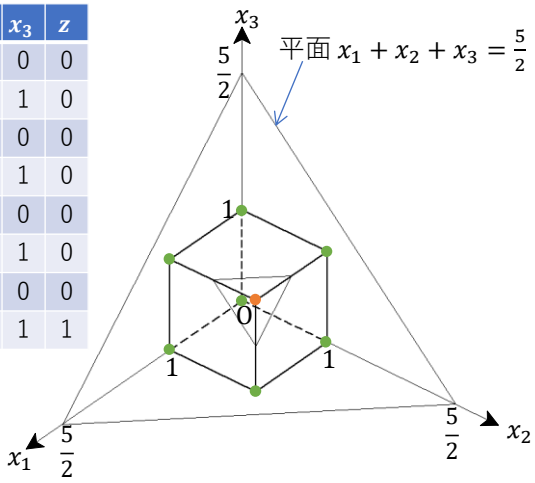
$w_1 = w_2 = w_3 = 1$
$\theta = \frac{5}{2}$

の場合,

$x_1 + x_2 + x_3 - \frac{5}{2} < 0 \rightarrow z = 0$

$x_1 + x_2 + x_3 - \frac{5}{2} \geqq 0 \rightarrow z = 1$

x_1	x_2	x_3	z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



ニューロセルの例 (3入力の場合(2))

$w_1 = w_2 = w_3 = 1$

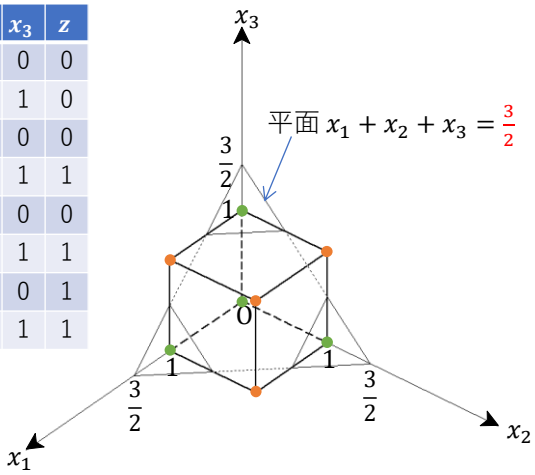
$\theta = \frac{3}{2}$

の場合,

$x_1 + x_2 + x_3 - \frac{3}{2} < 0 \rightarrow z = 0$

$x_1 + x_2 + x_3 - \frac{3}{2} \geqq 0 \rightarrow z = 1$

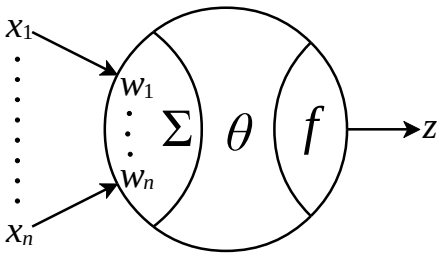
x_1	x_2	x_3	z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



重み(w_i)と閾値(θ)を変えれば境界の平面が変化する

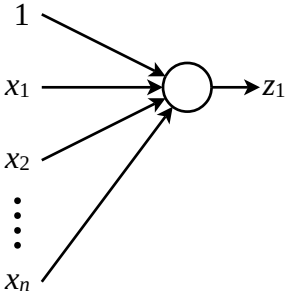
n入力に一般化して考えると……

- 各入力はn次元立方体の各頂点に対応する
- $w_1x_1 + \cdots + w_nx_n - \theta = 0$は,$n$次元空間内の$n-1$次元超平面に対応する
- 出力は，入力に対応する頂点が，上の超平面のどちら側にあるかで，0になるか1になるか定まる



n入力1出力単層ネットワーク

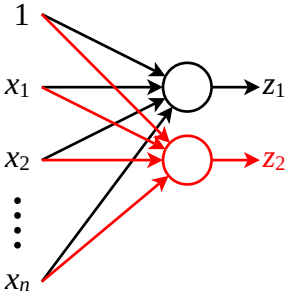
- 単一のセルと等価
  ＝重みと入力の内積 → 活性化関数



$$z_1 = f\left((w_{10}\ w_{11} \cdots w_{1n}) \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} \right)$$

n入力m出力単層ネットワーク

- 1出力のものをm個並列に並べる
  ＝重み行列と入力の積 → 活性化関数

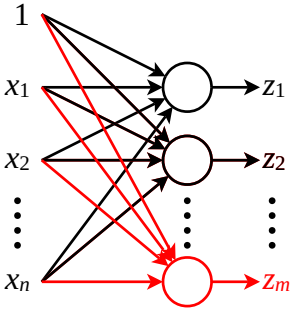


$$z_1 = f\left((w_{10}\ w_{11} \cdots w_{1n}) \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} \right)$$

$$z_2 = f\left((w_{20}\ w_{21} \cdots w_{2n}) \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} \right)$$

n入力m出力単層ネットワーク

- 1出力のものをm個並列に並べる
  ＝重み行列と入力の積 → 活性化関数



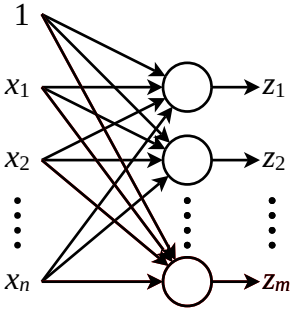
$$z_1 = f\left((w_{10}\ w_{11}\cdots w_{1n})\begin{pmatrix}1\\x_1\\\vdots\\x_n\end{pmatrix}\right)$$

$$z_2 = f\left((w_{20}\ w_{21}\cdots w_{2n})\begin{pmatrix}1\\x_1\\\vdots\\x_n\end{pmatrix}\right)$$

$\vdots$

n入力m出力単層ネットワーク

- 1出力のものをm個並列に並べる
  ＝重み行列と入力の積 → 活性化関数



$$\begin{pmatrix}1\\z_1\\\vdots\\z_m\end{pmatrix} = f\left(\begin{pmatrix}1 & 0 & \cdots & 0\\w_{10} & \cdots & & w_{1n}\\\vdots & & \ddots & \vdots\\w_{m0} & \cdots & & w_{mn}\end{pmatrix}\begin{pmatrix}1\\x_1\\\vdots\\x_n\end{pmatrix}\right)$$

$$\boldsymbol{z} = f(W\boldsymbol{x})$$

単層ネットワークの限界
（線形分離問題）

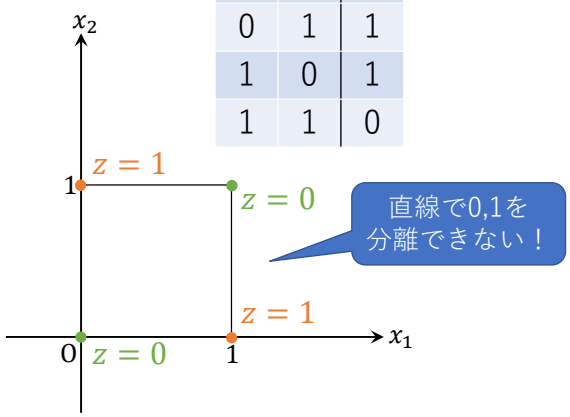
- 0になるべき入力と1になるべき入力を直線（超平面）で分けられない場合，単層のネットワークでは実現できない

↓

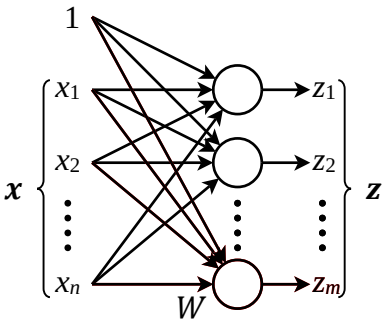
- 2層以上なら可能になる
- 多層になればなるほど，複雑な出力が生成できる

x_1	x_2	z
0	0	0
0	1	1
1	0	1
1	1	0

排他的論理和 (EXOR)



n入力m出力単層ネットワーク

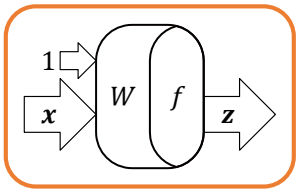


$$\begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} = \begin{pmatrix} w_{10} & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m0} & w_{m1} & \cdots & w_{mn} \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$$\boldsymbol{u} = \begin{pmatrix} \boldsymbol{w}_{*0} & W' \end{pmatrix} \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix}$$

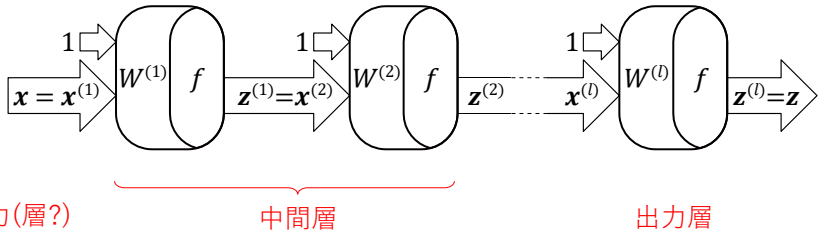
$$\boldsymbol{u} = W \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix}$$

$$\boldsymbol{z} = f(\boldsymbol{u})$$



フィードフォワードネットワーク (FNN)

- 単層ネットワークを複数直列に並べる

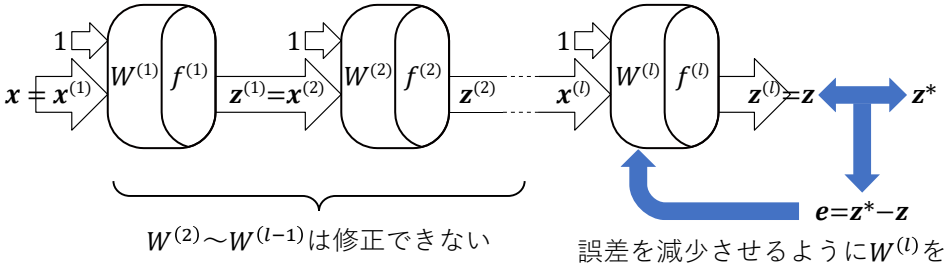


パーセプトロン（$=f$がステップ関数のFNN）の学習

- 訓練パターン（入力$\boldsymbol{x}$と正しい出力$\boldsymbol{z}^*$のペア）の集合を用意（教師あり学習）
- 各訓練パターンで、入力$\boldsymbol{x}$に対する出力信号$\boldsymbol{z}$と教師信号$\boldsymbol{z}^*$の誤差を縮めるように，出力層（最終層）の重み・バイアス（$W^{(l)}$）を修正
- 誤差が無くなるまで反復する

パーセプトロン（$=f$がステップ関数のFNN）の学習

- 訓練パターン$(\boldsymbol{x}, \boldsymbol{z}^*)$の入力信号$\boldsymbol{x}$に対する出力信号$\boldsymbol{z}$と教師信号$\boldsymbol{z}^*$の誤差$\boldsymbol{e}$を小さくするように，出力層の$W^{(k)}$（重みとバイアス）を調整する．



出力層の各ユニットごとに…

- $e = z^* - z$ の値が …

  0 … 問題なし
  ＋1 … 活性値が小さすぎる(増やす必要)
  －1 … 活性値が大きすぎる(減らす必要)

下2つの場合に1だった入力に対する重み（$x_0 = 1$としてバイアスw_0を含む）を増減する（0だった入力の重みは変えない）

$$w_i \to w_i + \Delta w_i, \quad \Delta w_i = \varepsilon e x_i \quad (i = 0 \sim n, \quad \varepsilon: \text{学習係数})$$

行ベクトルとしてまとめると…

$$\boxed{(w_0 \cdots w_n) \to (w_0 \cdots w_n) + \varepsilon e (x_0 \cdots x_n)}$$

$$\boxed{(w_0 \cdots w_n) \rightarrow (w_0 \cdots w_n) + \varepsilon e(x_0 \cdots x_n)}$$

複数出力（出力層がmユニット）においては …

• $\boldsymbol{e} = \begin{pmatrix} e_1 \\ \vdots \\ e_m \end{pmatrix} = \begin{pmatrix} {z^*}_1 \\ \vdots \\ {z^*}_m \end{pmatrix} - \begin{pmatrix} z_1 \\ \vdots \\ z_m \end{pmatrix} = \boldsymbol{z}^* - \boldsymbol{z}$ として,

$$\begin{pmatrix} w_{10} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m0} & \cdots & w_{mn} \end{pmatrix} \rightarrow \begin{pmatrix} w_{10} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m0} & \cdots & w_{mn} \end{pmatrix} + \varepsilon \begin{pmatrix} e_1 \\ \vdots \\ e_m \end{pmatrix} (x_0 \cdots x_n)$$

すなわち、

$$\boxed{W = W + \Delta W,\ \ \Delta W = \varepsilon \boldsymbol{e}\boldsymbol{x}^{\mathrm{T}},\ \ \boldsymbol{x}^{\mathrm{T}} = (x_0 \cdots x_n)}$$

$$\boxed{W = W + \Delta W,\ \ \Delta W = \varepsilon \boldsymbol{e}\boldsymbol{x}^{\mathrm{T}},\ \ \boldsymbol{x}^{\mathrm{T}} = (x_0 \cdots x_n)}$$

訓練パターンが複数（$(\boldsymbol{x}_1, {\boldsymbol{z}^*}_1) \ldots (\boldsymbol{x}_k, {\boldsymbol{z}^*}_k)$）の場合 …

$$X = (\boldsymbol{x}_1 \quad \ldots \quad \boldsymbol{x}_k)$$

$$E = (\boldsymbol{e}_1 \quad \ldots \quad \boldsymbol{e}_k) = ({\boldsymbol{z}^*}_1 \quad \ldots \quad {\boldsymbol{z}^*}_k) - (\boldsymbol{z}_1 \quad \ldots \quad \boldsymbol{z}_k) = Z - Z^*$$

とすると,

$$\sum_{i=1}^{k} \Delta W_i = \varepsilon \sum_{i=1}^{k} \boldsymbol{e}_i {\boldsymbol{x}_i}^{\mathrm{T}} = \varepsilon (\boldsymbol{e}_1 \quad \ldots \quad \boldsymbol{e}_k) \begin{pmatrix} {\boldsymbol{x}_1}^{\mathrm{T}} \\ \vdots \\ {\boldsymbol{x}_k}^{\mathrm{T}} \end{pmatrix} = \varepsilon E X^{\mathrm{T}}$$

したがって，複数の訓練パターンでまとめて学習する場合,

$$\boxed{W = W + \Delta W,\ \ \Delta W = \varepsilon E X^{\mathrm{T}}}$$