計算知能
(COMPUTATIONAL INTELLIGENCE)

第５回
ニューラルネットワークの学習アルゴリズム（１）
教員：　谷口彰

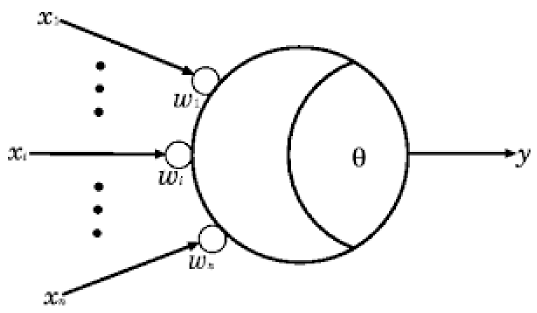
第5, 6回
ニューラルネットワークの学習アルゴリズム

- 単純パーセプトロン
  - 誤り訂正学習法
- 線形ニューラルネットワーク
  - デルタ則

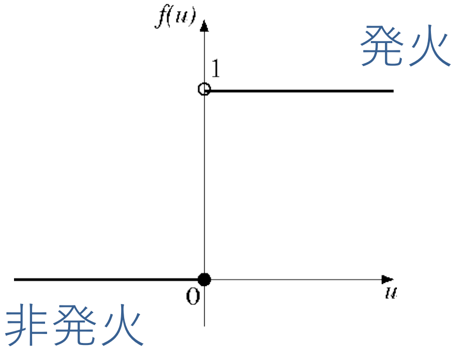
復習：ニューロンのモデル

$$\boxed{u_i = \sum_{j=1}^{n} w_{ij} x_j}$$

$$\boxed{y_i = f_i(u_i - \theta_i)}$$

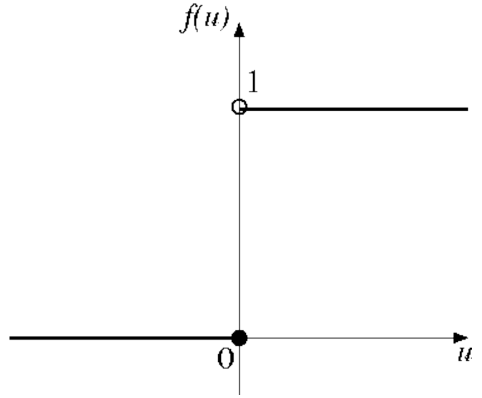


階段関数（単位ステップ関数、ヘビサイド関数）



復習：
しきい素子（階段関数）のモデル

f_iが階段関数のとき



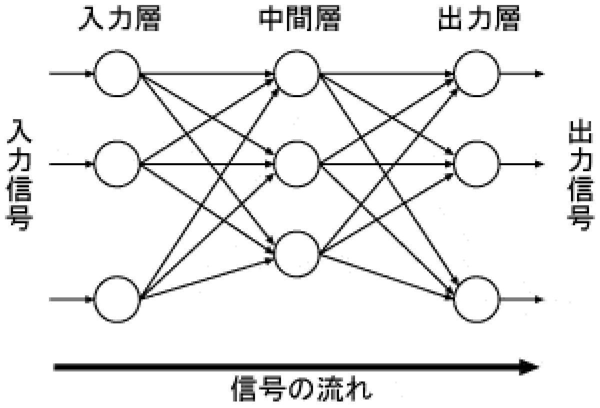
$$y_i = f_i(u_i - \theta_i) = \begin{cases} 1, & if \sum_{j=1}^{n} w_{ij}x_j - \theta_i \geq 0 \\ 0, & if \sum_{j=1}^{n} w_{ij}x_j - \theta_i < 0 \end{cases}$$

McCulloch-Pittsモデル：生体ニューロンを極めて単純化し、2つの本質的な特徴をとりいれた

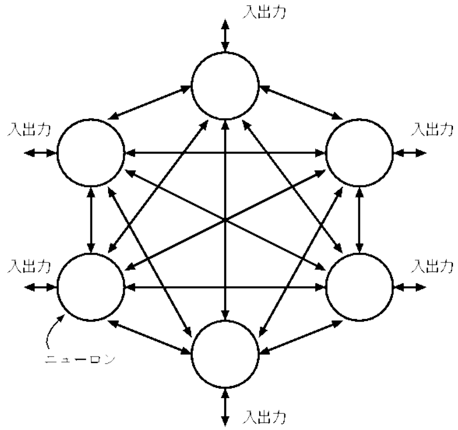
- ニューロンは、他の多数のニューロンからの信号を受け取り、その膜電位（内部状態）が定まる
- ニューロンは、その膜電位が閾値を超えなければ何もせず、超えれば興奮（発火）する

復習：
ニューロンのネットワーク

- ニューロンの様々な繋ぎ方でネットワークを構築



階層型ネットワーク



相互結合型ネットワーク

入力 ⇒ 出力

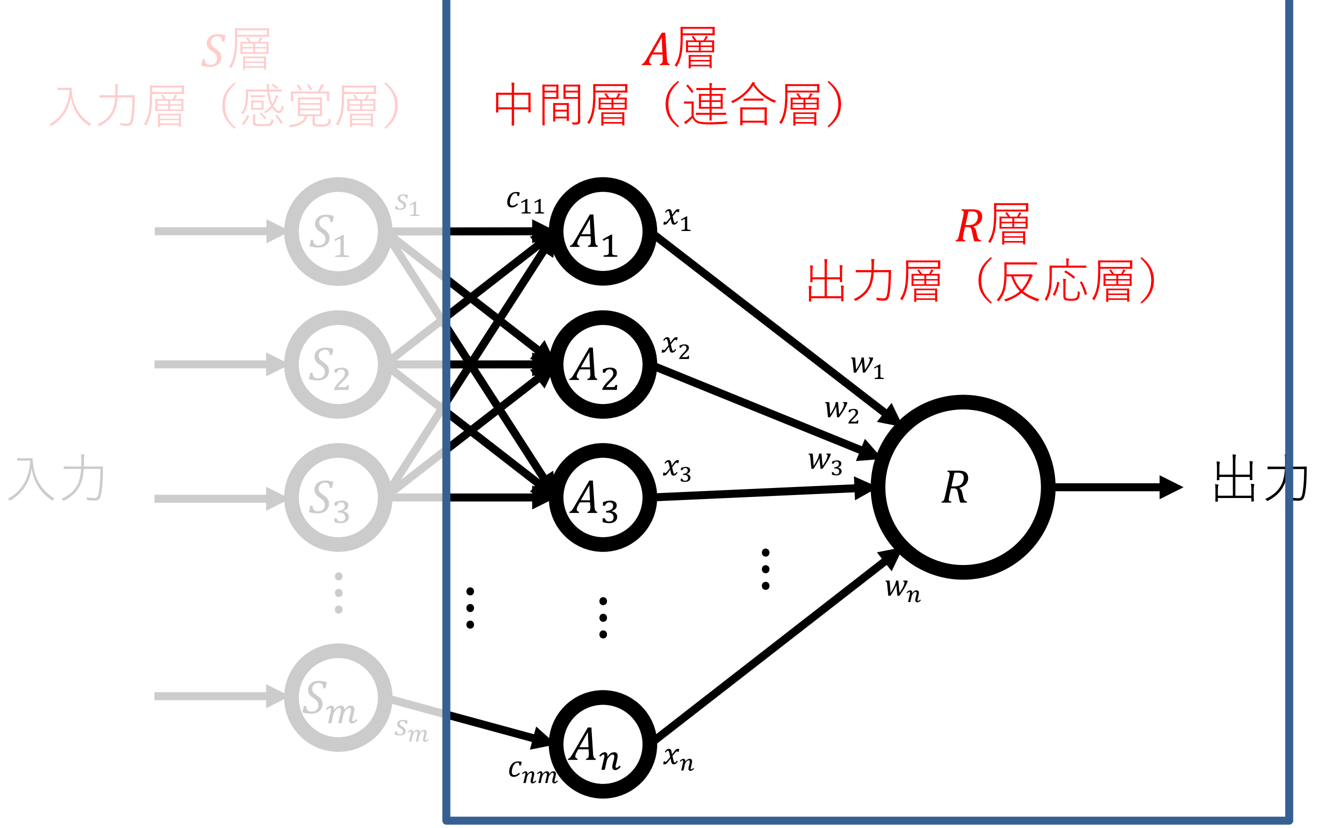
信号の流れは一方通行
（前進型処理）

初期：すべてのニューロンが何らかの状態

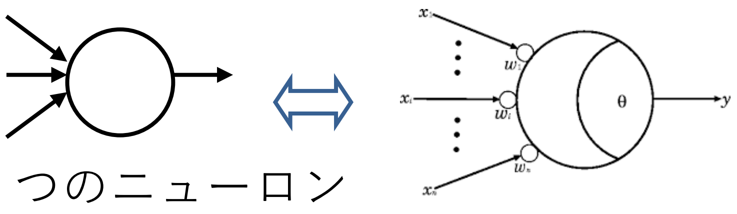
学習の経過

最終：収束状態

階層型ネットワークの出発点
単純パーセプトロン



フランク・ローゼンブラット（Frank Rosenblatt, 1928年7月11日 - 1971年7月11日）は、アメリカの心理学者。



単純パーセプトロン：二層で構成される形式ニューロンの階層型ネットワーク

- S層：外部から信号が与えられる。
- A層：S層からの情報に反応する。（結合荷重固定）
- R層：A層の答えに重みづけをして、多数決を行い、答えを出す。

単純パーセプトロン

可変な荷重w：学習

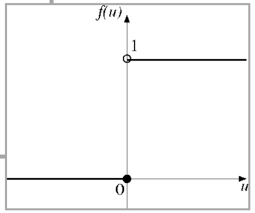
S層 → A層 → R層

S層
線形素子s_j
[線形関数]
単なる伝達
（mユニット）

A層
しきい素子x_i
[階段関数]
0, 1の二値
（nユニット）

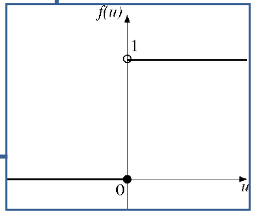
R層
しきい素子o
[階段関数]
0, 1の二値
（1ユニットのみ）

$$x_i = \begin{cases} 1, & \text{if } \sum_{j=1}^{m} c_{ij} s_j - \theta_i \geq 0 \\ 0, & \text{if } \sum_{j=1}^{m} c_{ij} s_j - \theta_i < 0 \end{cases}$$



A層のニューロン

$$o = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i x_i - \theta \geq 0 \\ 0, & \text{if } \sum_{i=1}^{n} w_i x_i - \theta < 0 \end{cases}$$

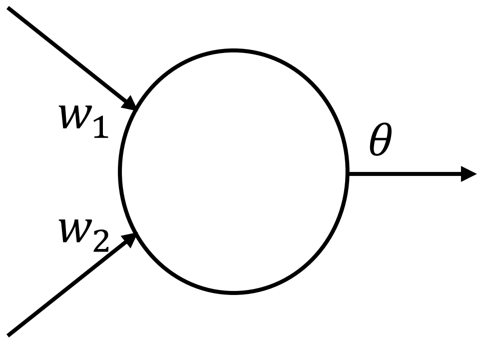


R層のニューロン

練習問題5-1

活性化関数は階段関数

- 以下の単純パーセプトロンを用いてAND回路を構築する際の重みw_iとしきい値θの値を一つ答えよ
ただし、各入力値は0もしくは1とする



単純パーセプトロンの学習

誤り訂正学習法（教師あり学習）：

出力値と教師信号が一致しない「誤り」の場合のみ、結合荷重と閾値が修正される

修正量 $\boxed{\Delta \mathbf{w} = \eta \cdot (t - o) \cdot \mathbf{x}}$

η：学習率（小さな正値）

$\mathbf{w}$: A層とR層の間の結合荷重と閾値

$\mathbf{x}$: A層のn個のユニットの出力値、0,1の二値

o: R層のユニットの出力値、0,1の二値

t: 教師信号、0,1の二値

修正後の$\mathbf{w}$ = 修正前の$\mathbf{w}$ + 修正量$\Delta\mathbf{w}$

単純パーセプトロンの学習

$$o = \begin{cases} 1, & if \ \sum_{j=1}^{n} w_{ij}x_j - \theta_i \geq 0 \\ 0, & if \ \sum_{j=1}^{n} w_{ij}x_j - \theta_i < 0 \end{cases} \quad (1)$$

ここで、記号の簡単化のために、常に1の値をとる(n+1)番目の入力を仮定し、対応する結合荷重を $w_{n+1} = -\theta$ と考え、(n+1)次元の2つのベクトルを導入

$$\mathbf{x} = (x_1, x_2, \cdots, x_n, 1), \mathbf{w} = (w_1, w_2, \cdots, w_n, w_{n+1}) \quad (2)$$

単純パーセプトロンの学習

閾値θをn+1番目の結合荷重$w_{n+1} = -\theta$に置き換える

このとき、式(1)は式(3)の様に簡潔に表すことができる

$$o = \begin{cases} 1, & if\ \mathbf{wx} \geq 0 \\ 0, & if\ \mathbf{wx} < 0 \end{cases} \tag{3}$$

ここで $\mathbf{wx}$ は2つのベクトル $\mathbf{w}$ と $\mathbf{x}$ の内積、すなわち

$$\sum_{j=1}^{n+1} w_{ij} x_j$$

を表す

誤り訂正学習法

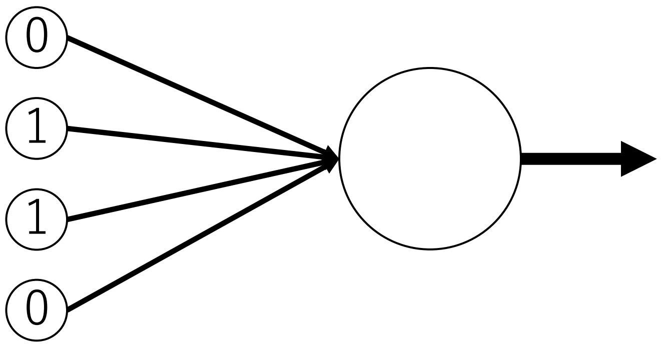
出力値 o と教師信号 t が一致せず、$x_i = 1$のときのみ、w_i が修正される

$$\boxed{\Delta \mathbf{w} = \eta \cdot (t - o) \cdot \mathbf{x}}$$

η：学習率
（小さな正値）

$\mathbf{w}$：A層とR層間の結合重み　$\mathbf{x}$：A層のn個のユニットの出力値（0,1）

o：R層のユニットの出力値（0,1）　t：教師信号（0,1）



if

$o \neq t$

$o < t,\ t - o = 1$　→　wはηだけ増加

$o > t,\ t - o = -1$　→　wはηだけ減少

反応すべきなのに
反応しなかった。

反応してはいけないのに
反応してしまった。

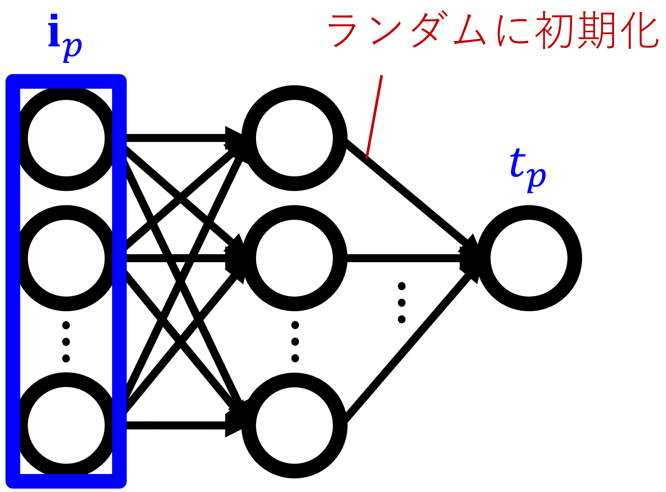
出力値 o が教師信号 t に近づくように修正される

5 分休憩

単純パーセプトロンの学習アルゴリズム(1/5)

1. 入力パターンベクトル $\mathbf{i}_p = \left(i_{p1}, \cdots, i_{pm}\right)$ と対応する教師信号 t_p の組の集合 $(p = 1, \dots, P)$を考える

2. 結合荷重 $\mathbf{w} = (w_1, \dots, w_n, w_{n+1})$ の初期値をランダムに小さな値に設定する
   さらに、学習率 $\eta (0 < \eta \leq 1)$ を設定する

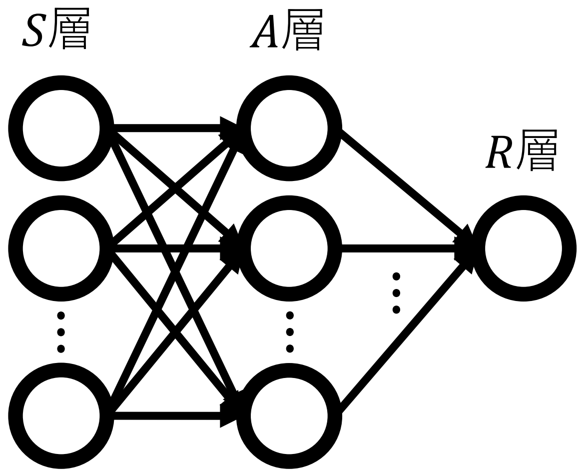


単純パーセプトロンの学習アルゴリズム(2/5)

3. P個の入力パターンベクトルの中の１つの入力パターンベクトル $\mathbf{i}_p = \left(i_{p1}, \dots, i_{pm}\right)$ を選び、 $\mathbf{i}_p$ に対するA層の出力 $x_j (j = 1, \dots, n)$ を

$$x_j = \begin{cases} 1, \sum_{i=1}^{m} c_{ji} i_{pi} - \theta_j \geq 0 & \text{のとき} \\ 0, \sum_{i=1}^{m} c_{ji} i_{pi} - \theta_j < 0 & \text{のとき} \end{cases}$$

により計算する

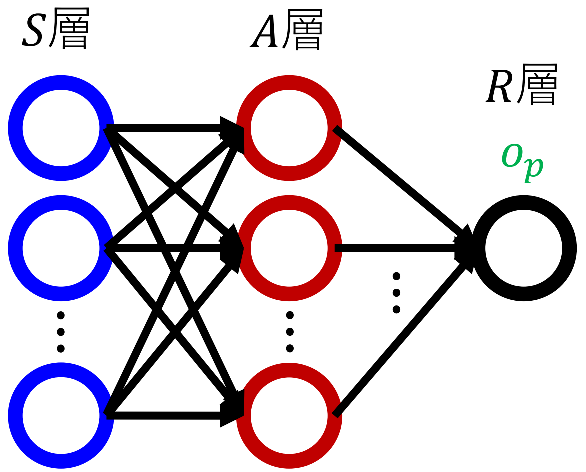


単純パーセプトロンの学習アルゴリズム(3/5)

4. 得られたA層の出力ベクトル $\mathbf{x}_p = \left(x_{p1}, \ldots, x_{pn}, 1\right)$ からR層のユニットRの出力 o_p を

$$o_p = \begin{cases} 1, \mathbf{wx}_p \geq 0 & \text{のとき} \\ 0, \mathbf{wx}_p < 0 & \text{のとき} \end{cases}$$

により求める

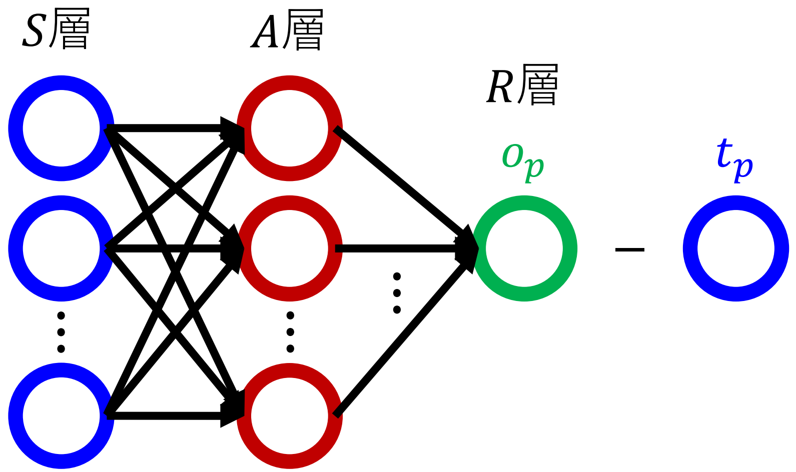


単純パーセプトロンの学習アルゴリズム(4/5)

5. 得られた出力値 o_p と教師信号 t_p との誤差 $t_p - o_p$ を用いて、結合荷重ベクトルを

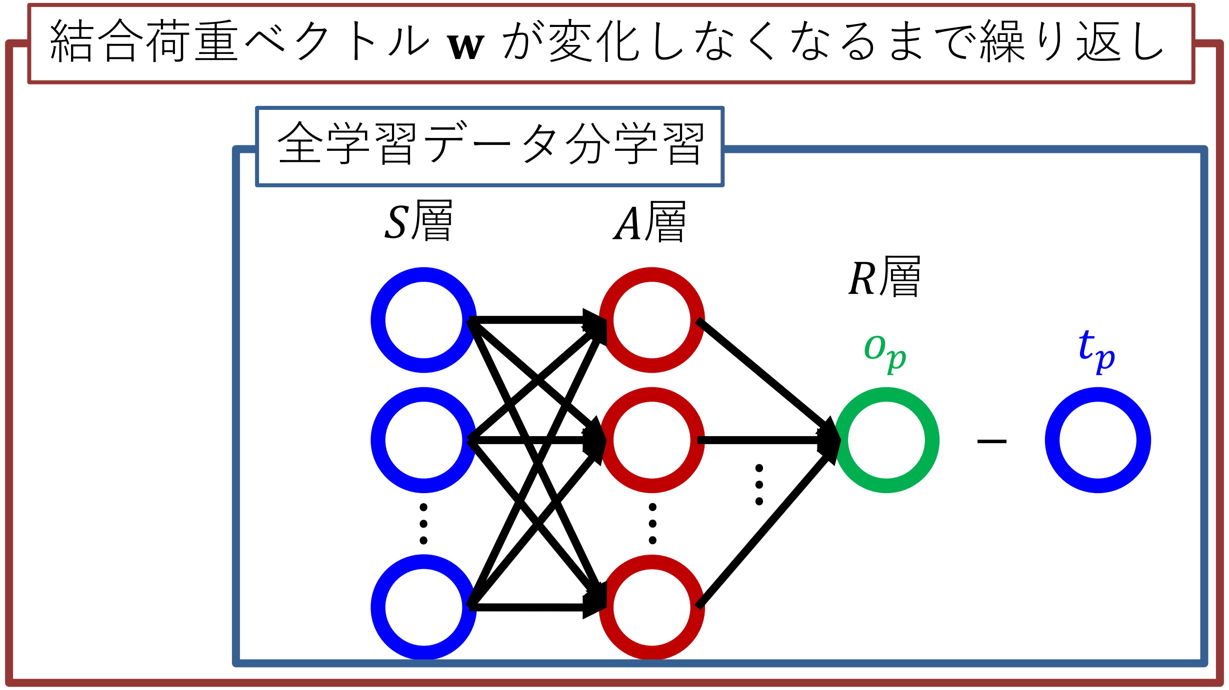
$$\mathbf{w} \leftarrow \mathbf{w} + \eta(t_p - o_p)\mathbf{x}_p \qquad 0 < \eta \leq 1$$

により修正する



単純パーセプトロンの学習アルゴリズム(5/5)

6. すべての入力パターンベクトル $\mathbf{i}_p = \left(i_{p1}, \dots, i_{pm}\right)$ $(p = 1, \dots, P)$ に対して、結合荷重ベクトル $\mathbf{w}$ が変化しなくなれば終了する
そうでなければ、手順3から手順5までの操作を繰り返す



単純パーセプトロンは何をしているか

$$o = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i x_i - \theta \geq 0 \\ 0, & \text{if } \sum_{i=1}^{n} w_i x_i - \theta < 0 \end{cases}$$

つまり

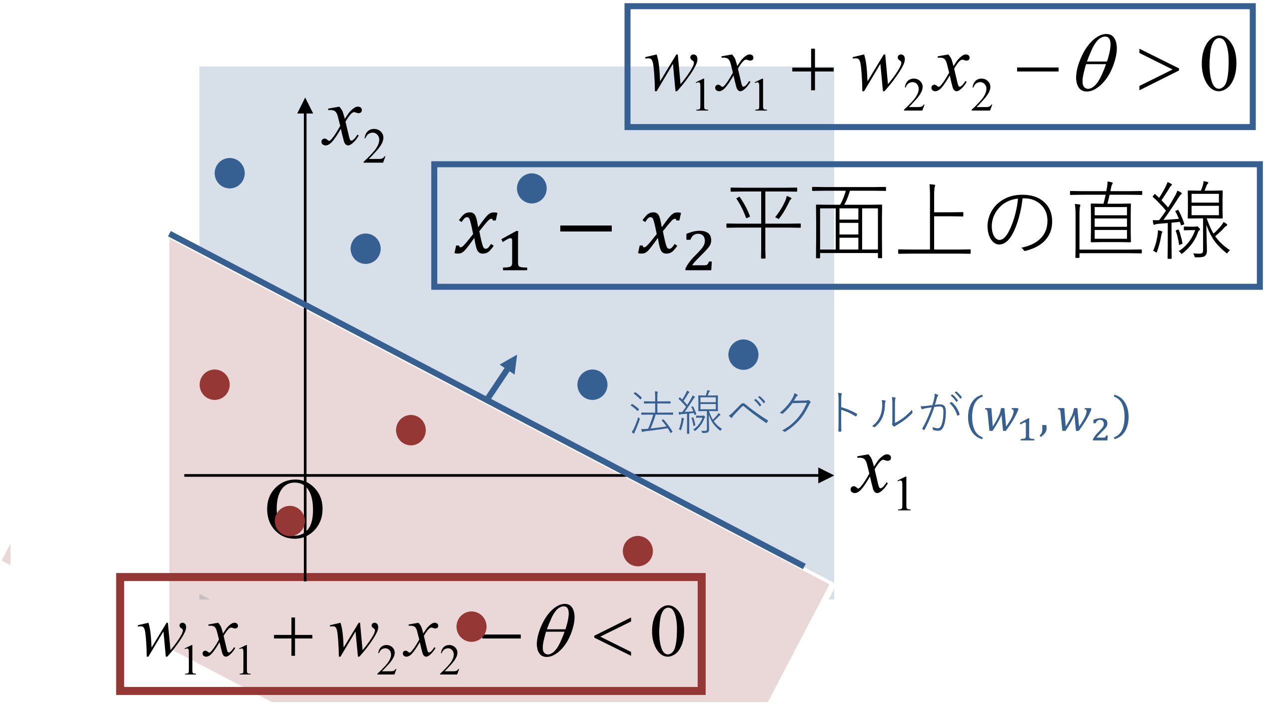
$\sum_{i=1}^{n} w_i x_i - \theta = 0$　あるいは　$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n - \theta = 0$

が、1か0かの「境界」となっている

この式は何を表しているか

$\boxed{w_1x_1 + w_2x_2 + \ldots + w_nx_n - \theta = 0}$ とは？

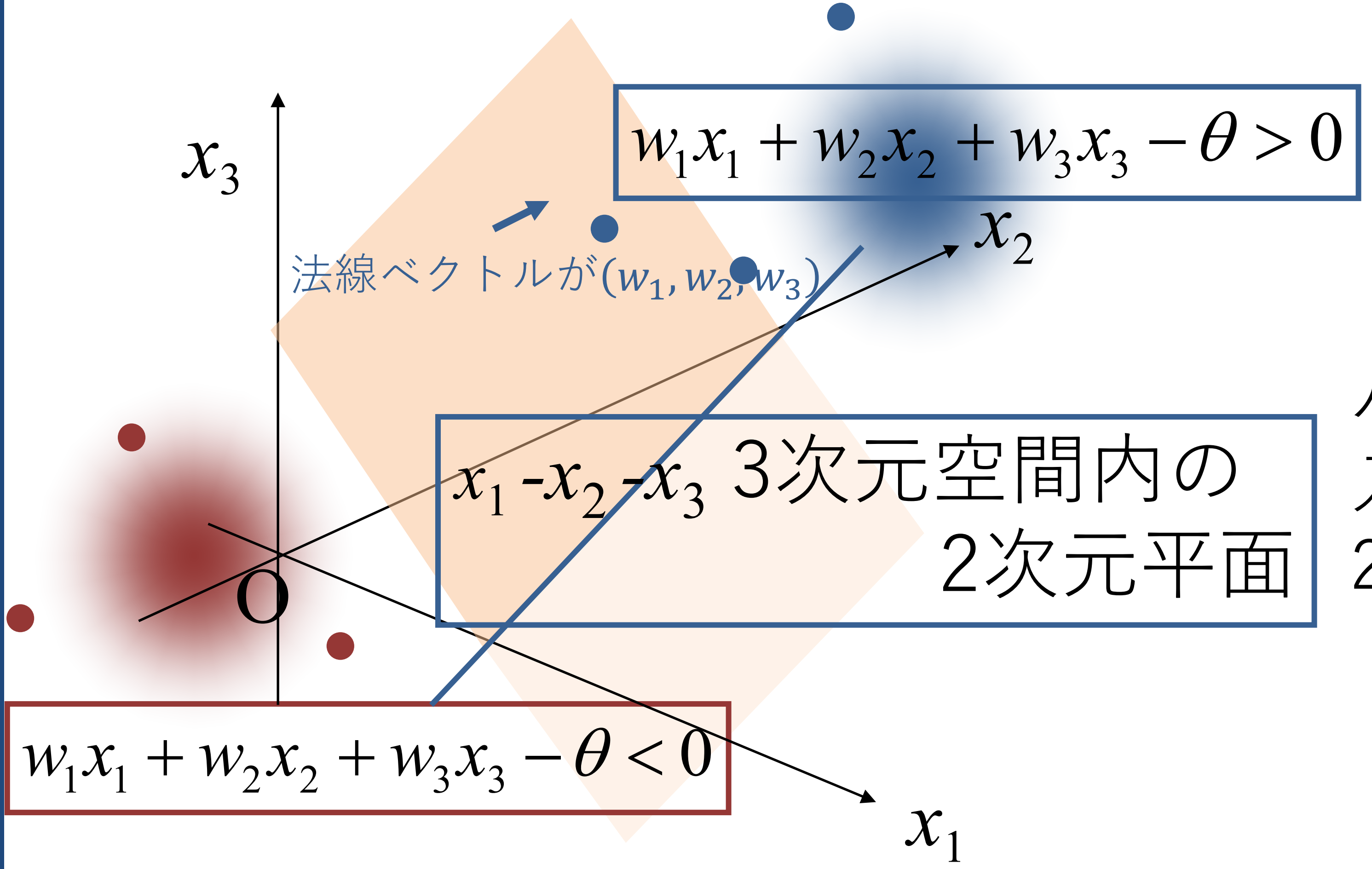
n=2 のとき $\boxed{w_1x_1 + w_2x_2 - \theta = 0}$ とは、



パーセプトロンは、
入力値 ●(x_1, x_2)に応じて
2種類の答えを出力

$$o = \begin{cases} 1, & \text{if } \sum_{i=1}^{2} w_i x_i - \theta \geq 0 \\ 0, & \text{if } \sum_{i=1}^{2} w_i x_i - \theta < 0 \end{cases}$$

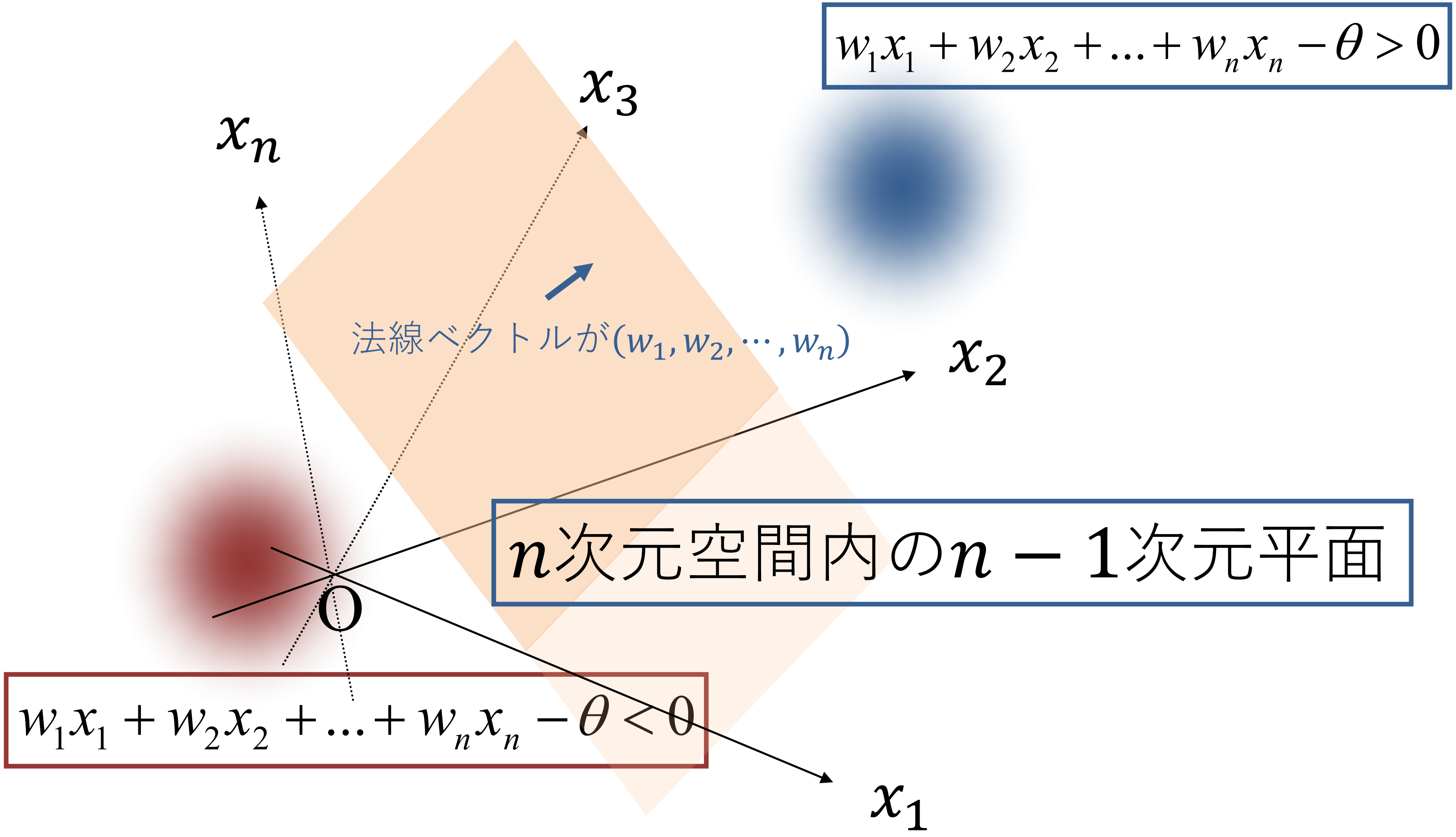
n=3 のとき $\boxed{w_1x_1 + w_2x_2 + w_3x_3 - \theta = 0}$ とは、



パーセプトロンは,
入力値 ●(x_1, x_2, x_3)に応じて
2種類の答えを出力

$$o = \begin{cases} 1, & \text{if } \sum_{i=1}^{3} w_i x_i - \theta \geq 0 \\ 0, & \text{if } \sum_{i=1}^{3} w_i x_i - \theta < 0 \end{cases}$$

一般のnのとき $\boxed{w_1x_1 + w_2x_2 + \ldots + w_nx_n - \theta = 0}$ とは、



単純パーセプトロンは何をしているか

$$\boxed{o = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i x_i - \theta \geq 0 \\ 0, & \text{if } \sum_{i=1}^{n} w_i x_i - \theta < 0 \end{cases}}$$

つまり

$\sum_{i=1}^{n} w_i x_i - \theta = 0$　あるいは　$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n - \theta = 0$

というn次元空間内の$n-1$次元平面

によってn次元空間（内の点）を2つに分離している

線形分離可能な2グループ

n次元の複数のデータ $(x_1, x_2, \dots, x_n)$ が、
２つのグループX^+、 X^-に分類されているとする

もし、
X^+に属するデータ（●点）とX^-に属するデータ（●点）を
分離する$n-1$次元（超）平面があるとき、
X^+とX^-は線形分離可能であるという

$\boxed{x_1, x_2, \dots, x_n\text{の一次（線形）式で表される}}$

単純パーセプトロンは何をしているか

- 線形分離可能な2クラスのデータに対しては、適切なパラメータ（結合荷重$w_1, w_2, \ldots, w_n$とθ）の値を用いることで、

  一方のクラスに属するデータに対しては出力0を
  他方のクラスに属するデータに対しては出力1を

  返すことができる → 全てのデータを正しく分類できる

- 更に、誤り訂正学習法を用いることで、有限回の学習回数（訂正を行う繰り返し回数）で適切なパラメータ値を獲得できることが証明されている
  【パーセプトロンの収束定理】

→ 線形分離可能な2クラス分類課題に対しては、それを実現する単純パーセプトロンが学習によって必ず得られる

まとめ

- 階層型ネットワークの始まりである単純パーセプトロンについて説明した。
- 単純パーセプトロンの結合荷重と閾値を学習する誤り訂正学習法について説明した。
- 単純パーセプトロンが何をしているか、数学的に理解した。

復習問題

1. 単純パーセプトロンで学習するパラメータは何か？

2. 誤り訂正学習法における「誤り」とは何か？

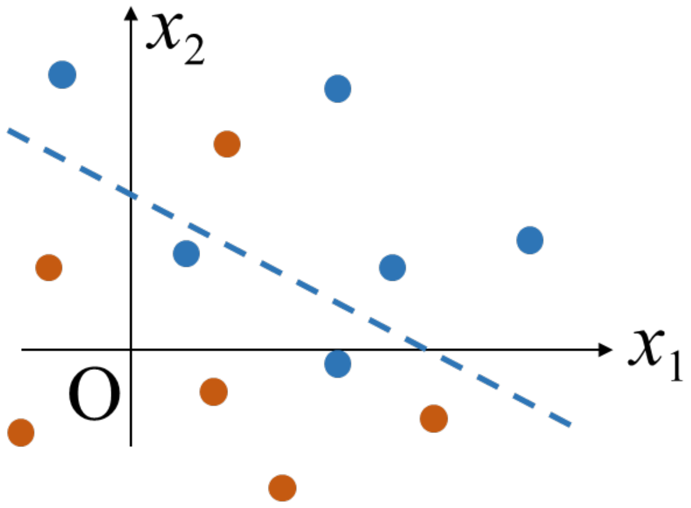
3. 単純パーセプトロンが解けるのはどのような課題か？

4. 単純パーセプトロンのR層に用いられる活性化関数は何か？

次回の講義

- ニューラルネットワークの学習アルゴリズム
  - 線形ニューラルネットワーク
  - デルタ則

線形分離不可能な2次元データ



どのような考えで
分離するか

計算知能
(COMPUTATIONAL INTELLIGENCE)

第６回
ニューラルネットワークの学習アルゴリズム（２）
教員：　谷口彰

復習：単純パーセプトロンは何をしているか

$$o = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i x_i - \theta \geq 0 \\ 0, & \text{if } \sum_{i=1}^{n} w_i x_i - \theta < 0 \end{cases}$$

つまり

$\sum_{i=1}^{n} w_i x_i - \theta = 0$　あるいは　$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n - \theta = 0$

というn次元空間内の$n-1$次元平面

によってn次元空間（内の点）を2つに分離している

復習：線形分離可能な2グループ

n次元の複数のデータ $(x_1, x_2, \ldots, x_n)$ が、２つのグループX^+、X^-に分類されているとする

もし、X^+に属するデータ（●点）とX^-に属するデータ（●点）を分離する$n-1$次元（超）平面があるとき、X^+とX^-は線形分離可能であるという

$x_1, x_2, \ldots, x_n$の一次（線形）式で表される

復習：単純パーセプトロンは何をしているか

- 線形分離可能な2クラスのデータに対しては、適切なパラメータ（結合荷重$w_1, w_2, \ldots, w_n$とθ）の値を用いることで、

  一方のクラスに属するデータに対しては出力0を

  他方のクラスに属するデータに対しては出力1を

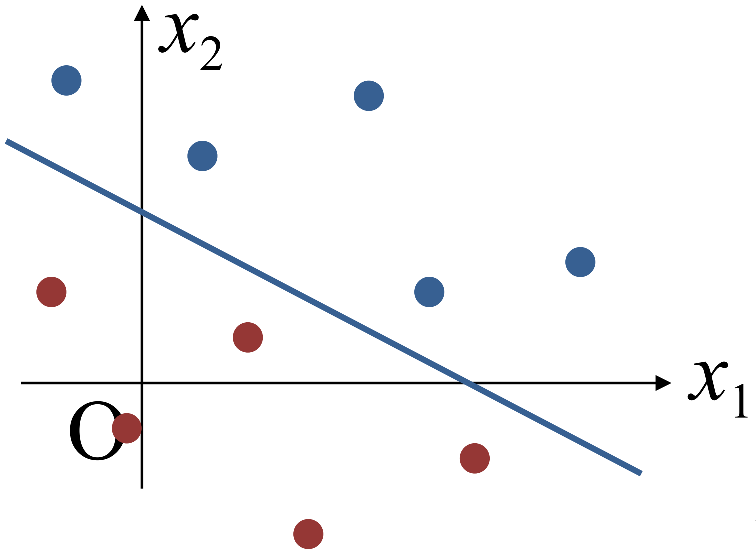
  返すことができる → 全てのデータを正しく分類できる

- 更に、誤り訂正学習法を用いることで、有限回の学習回数（訂正を行う繰り返し回数）で適切なパラメータ値を獲得できることが証明されている

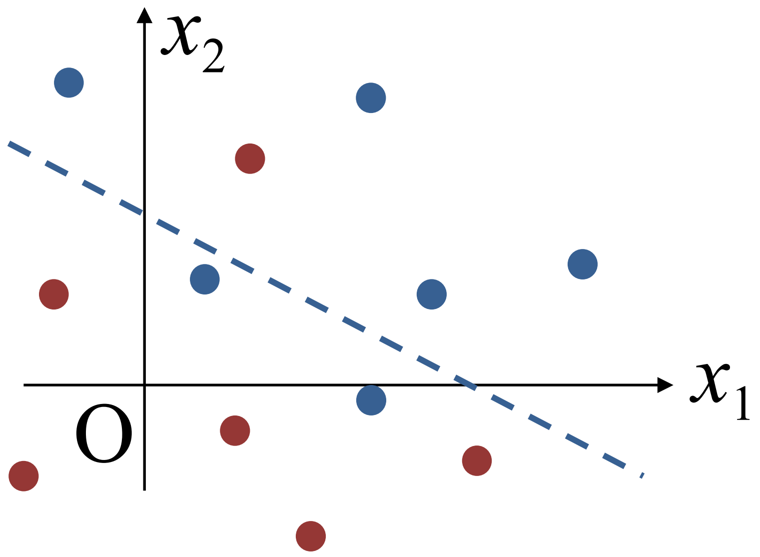
  【パーセプトロンの収束定理】

→ 線形分離可能な2クラス分類課題に対しては、それを実現する単純パーセプトロンが学習によって必ず得られる

線形分離可能な2次元データ



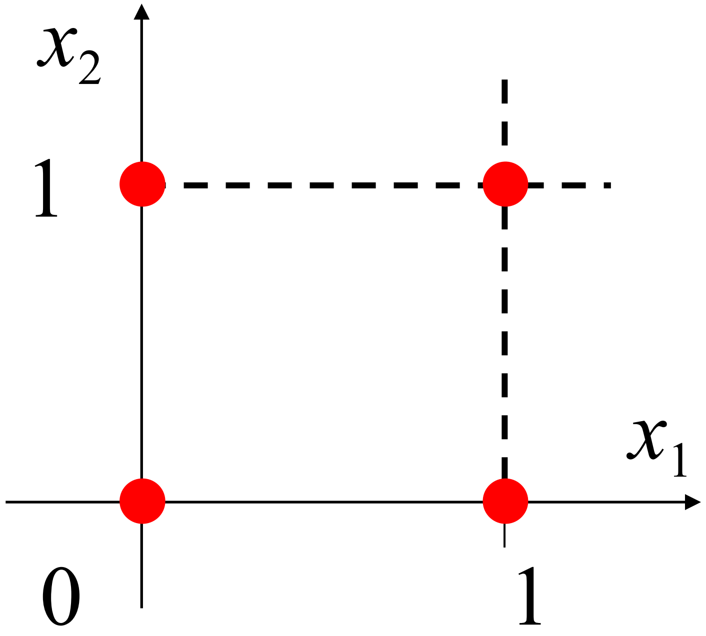
線形分離不可能な2次元データ



どのような考えで
分離するか

練習問題5-2

■ 2入力の論理関数を考える



x_1	x_2	AND	OR	XOR
0	0	?	?	?
0	1	?	?	?
1	0	?	?	?
1	1	?	?	?

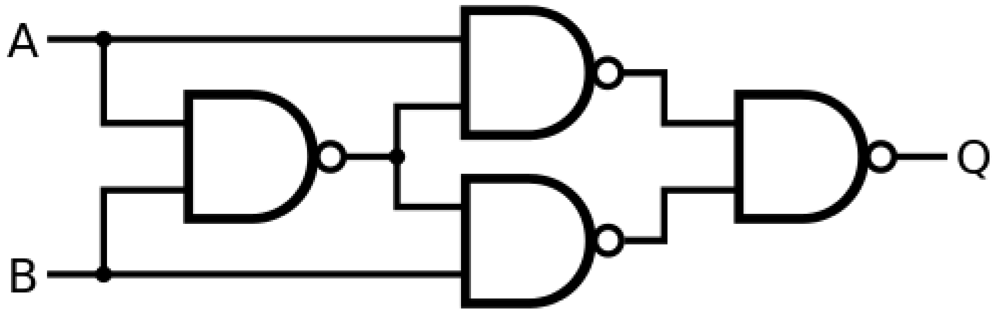
- ? は それぞれ 0か1か
- 各関数（AND, OR, XOR）は線形分離可能か

NANDゲートで構成したXORゲート

- XORゲートは線形分離可能なNANDゲートやNORゲートの組み合わせで構成可能

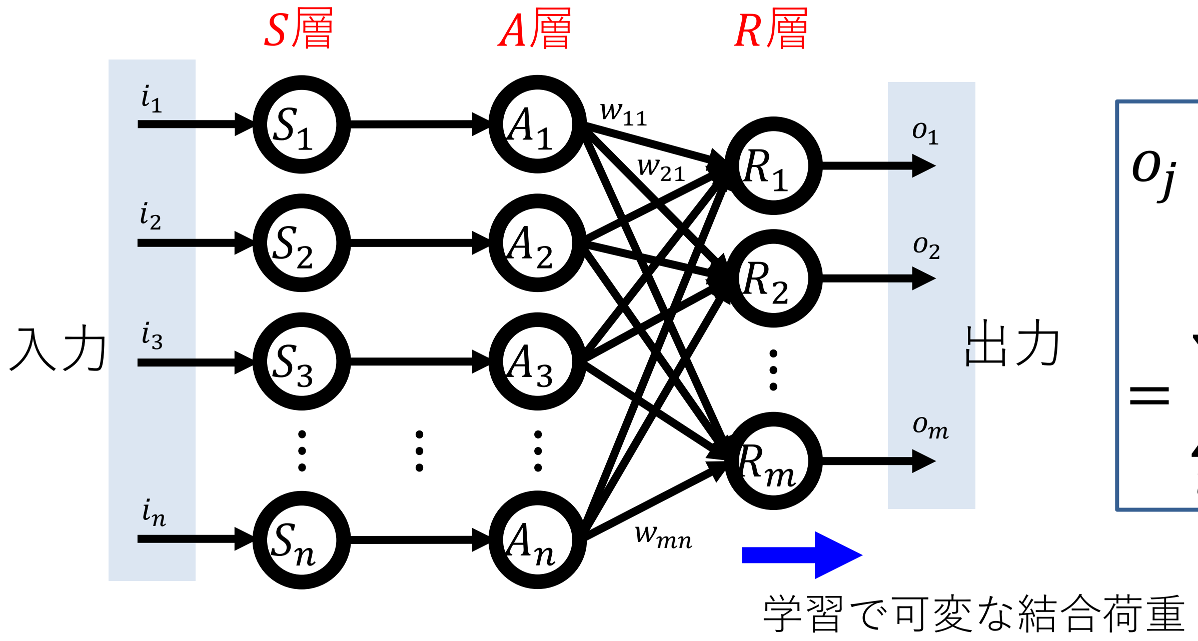
⇩

- 単純パーセプトロンにおいて線形分離不可能な問題は階層を深くすることにより分離可能となる

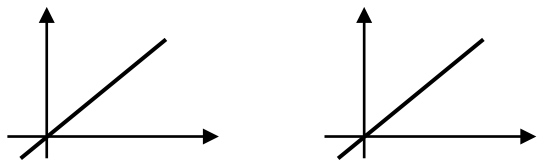


NANDゲートのみで構成したXORゲート

線形ニューロンの2層ネットワーク



$$o_j = f(u_j) = u_j = \sum_{i=1}^{n} w_{ji} o_i$$



$f(x) = x$：線形の関数（何も変換しない）

ネットワークの入出力関係と学習

- 入力 **oi** と出力 **oj**

$$o_j = \sum_{i=1}^{n} w_{ji} o_i$$

- 誤り訂正学習

$$\Delta\mathbf{w} = \eta \cdot (t - o) \cdot \mathbf{x}$$

$$\Rightarrow \boxed{\Delta w_{ji} = \eta\left(\boxed{t_j - o_j}\right) o_i}$$

出力ユニット j ごとに修正

$\boxed{\delta}$

δ、Δ、d：差、defference

デルタ則（Widrow-Hoff則）

別称：Δルール（教師あり学習）

連続値をとるA層とR層の2層からなるネットワークに、

誤り訂正学習法を拡張して適用

$$\boxed{\Delta w_{ji} = \eta \cdot \left(t_j - o_j\right) \cdot o_i}$$

η：学習率（小さな正値）

w_{ji}：A層の第iユニットからR層間の第jユニットへの結合荷重

o_i：A層の第iユニットの出力値、連続値（$i = 1, \ldots, n$）

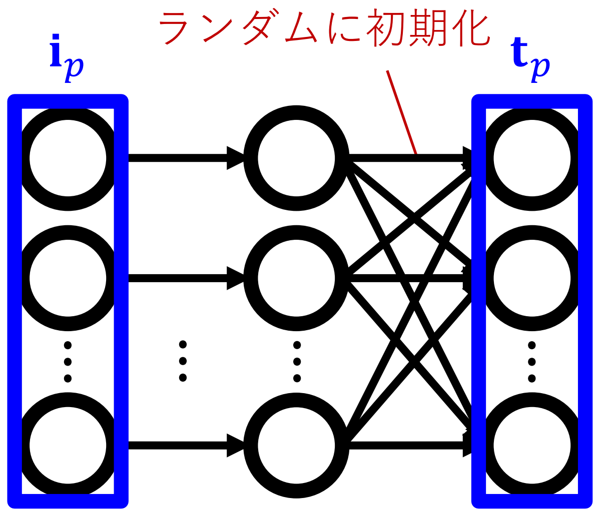
o_j：R層の第jユニットの出力値、連続値（$j = 1, \ldots, m$）

t_j：R層の第jユニットに対する教師信号、連続値

デルタ則のアルゴリズム(1/4)

1. 入力パターンベクトル $\mathbf{i}_p = (i_{p1}, \ldots, i_{pn})$ と対応する教師信号ベクトル $\mathbf{t}_p = (t_{p1}, \ldots, t_{pm})$ の組の集合（$p = 1, \ldots, P$）を考える

2. 結合荷重 w_{ji} （ $i = 1, \ldots, n \quad j = 1, \ldots, m$ ）の初期値をランダムに小さな値に設定する
   さらに、学習率 $\eta (0 < \eta \leq 1)$ を設定する

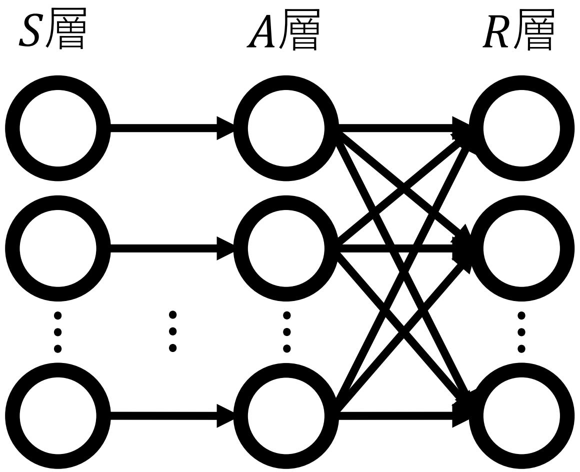


デルタ則のアルゴリズム(2/4)

3. P個の入力パターンベクトルの中の１つの入力パターンベクトル $\mathbf{i}_p = \left(i_{p1}, \dots, i_{pn}\right)$ を選び、 $\mathbf{i}_p$ に対するR層の各ユニット R_j $(j = 1, \dots, m)$ の出力

$$o_{pj} = \sum_{i=1}^{n} w_{ji} i_{pi} \qquad (j = 1, \dots, m)$$

を計算する

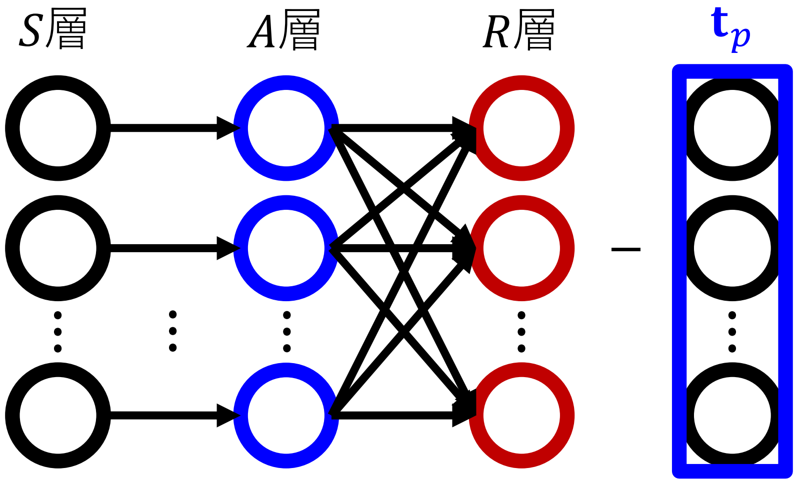


デルタ則のアルゴリズム(3/4)

4. 得られた出力 o_{pj} と教師信号 t_{pj} との
   誤差 $\delta_{pj} = t_{pj} - o_{pj}$ （$j = 1, \dots, m$） を用いて、
   結合荷重 w_{ji} （ $i = 1, \dots, n \quad j = 1, \dots, m$ ） をデルタ則

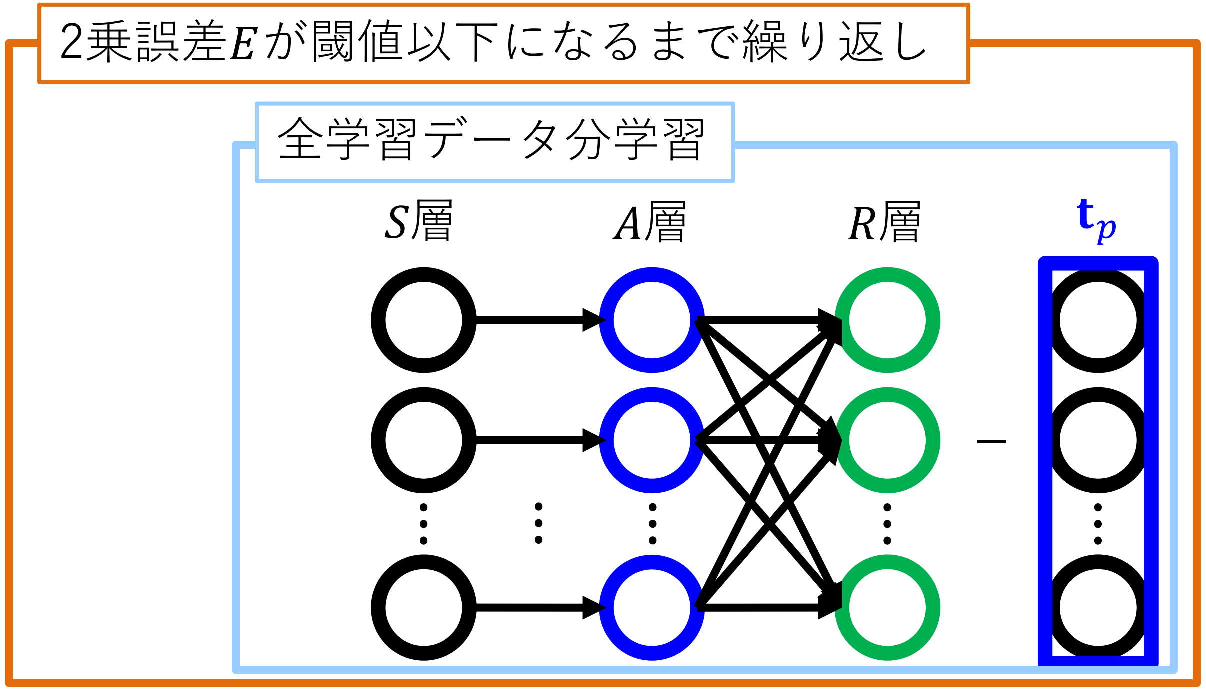
$$w_{ji} \leftarrow w_{ji} + \eta \delta_{pj} o_{pi}$$

   により修正する



デルタ則のアルゴリズム(4/4)

5. すべての入力パターンベクトル $\mathbf{i}_p = (i_{p1}, \ldots, i_{pn})$ $(p = 1, \ldots, P)$ に対する 2乗誤差 E が設定値以下になれば学習が収束したと判断して終了する そうでなければ、手順3、4の操作を繰り返す



デルタ則は最急降下法

最急降下法：関数$E(\mathbf{w})$の最小点（実際は極小点）を探す方法

$$\boxed{\Delta\mathbf{w} = -\eta \cdot \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} \propto -\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}}$$

ここで、A層-R層の2層のネットワークに対する出力誤差の2乗和

誤差（の総和） $$\boxed{E(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^{m}\left(t_j - o_j\right)^2}$$

を考えると、デルタ則は最急降下法となっている

デルタ則

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \eta \left({\partial f(\mathbf{x})}/{\partial \mathbf{x}} \right)$$

復習：最急降下法

$$= \mathbf{x}^{(k)} - \eta \begin{bmatrix} {\partial f(\mathbf{x})}/{\partial x_1} \\ {\partial f(\mathbf{x})}/{\partial x_2} \\ \vdots \\ {\partial f(\mathbf{x})}/{\partial x_n} \end{bmatrix}$$

最急降下法

$$\Delta w_{ji} = -\eta \cdot \frac{\partial E(\mathbf{w})}{\partial w_{ji}}$$

$$= \eta \cdot \delta_j \cdot o_i$$

誤差

$$E(\mathbf{w}) = \frac{1}{2} \sum_{j=1}^{m} (t_j - o_j)^2$$

最小化したい$\mathbf{w}$の関数E

$$\frac{\partial E(\mathbf{w})}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \cdot \frac{1}{2} \sum_{j=1}^{m} \left(t_j - o_j(\mathbf{w})\right)^2$$

$$\rightarrow \frac{\partial}{\partial w_{ji}} \cdot \frac{1}{2} \left(t_j - o_j(w_{ji})\right)^2$$

$$= \left(t_j - o_j(w_{ji})\right) \cdot (-1) \cdot \frac{\partial o_j}{\partial w_{ji}}$$

$$= -(t_j - o_j) \cdot o_i$$

$$= -\delta_j \cdot o_i$$

合成関数の微分

$$o_j = \sum_{i=1}^{n} w_{ji} o_i$$

合成関数の微分

$y = f(x)$、$z = g(y)$のとき、$z = g\big(f(x)\big) = g \circ f(x)$

z を x で微分するには

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x} = \frac{\partial g(y)}{\partial y} \cdot \frac{\partial f(x)}{\partial x} = g'(y) \cdot f'(x)$$

練習問題6-1

$\frac{1}{2}(1-x^2)^2$ を、合成関数の微分法を利用して微分しなさい

$-\dfrac{\partial E(\mathbf{w})}{\partial \mathbf{w}}$が最急降下方向であること(1/2)

一般に、 $\mathbf{x}$ の関数 $f(\mathbf{x})$ の最小値（実際には極小値）を見つける方法

最急降下：最も急速に f の値が小さくなる $\mathbf{x}$ の方向は

$$\boxed{-\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \left[-\frac{\partial f(\mathbf{x})}{\partial x_1}, -\frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, -\frac{\partial f(\mathbf{x})}{\partial x_n}\right]^T = -grad\, f(\mathbf{x})}$$

となる

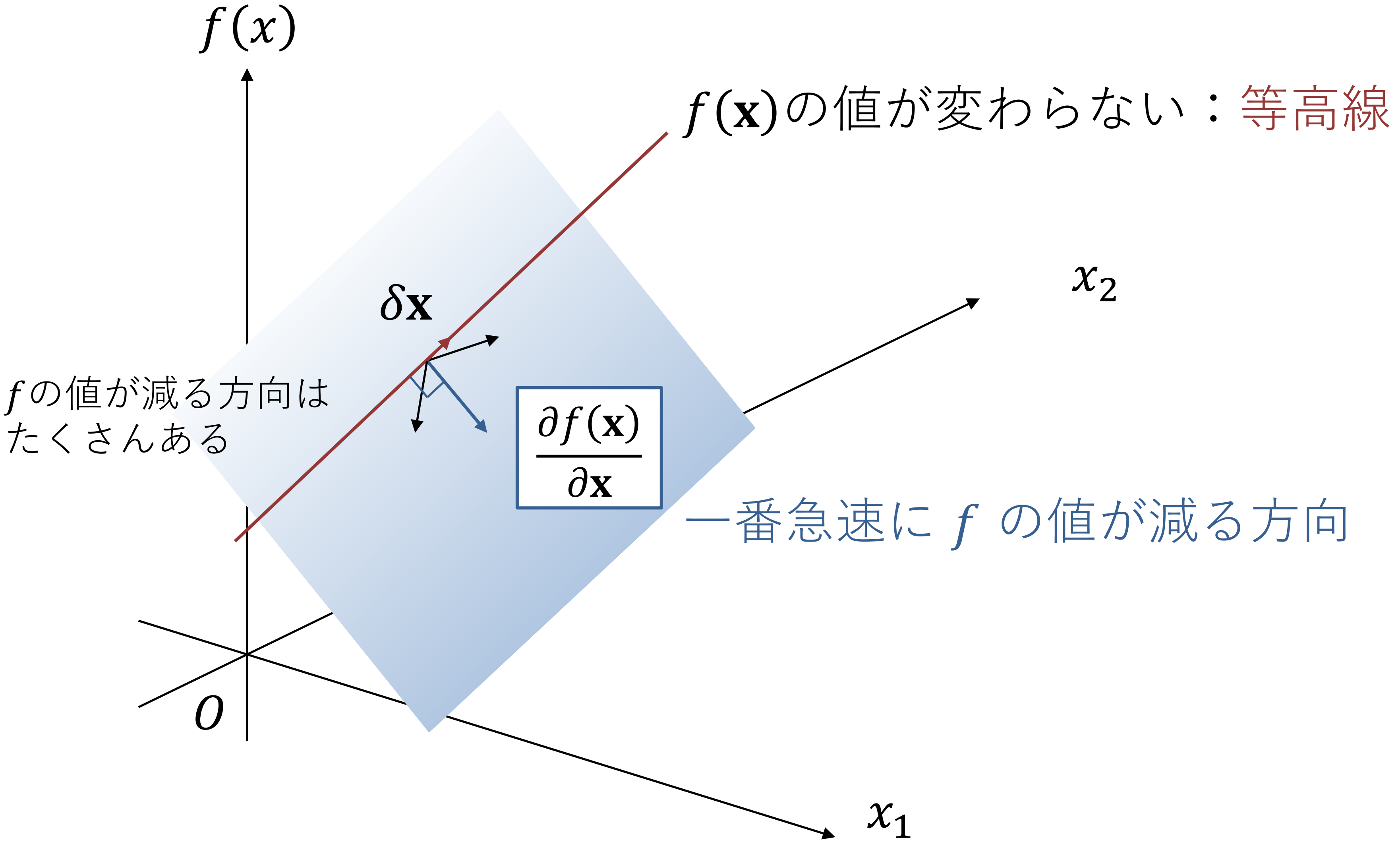
これは、 f が変化しない x の方向（ f の等高線に沿った向き）と、

$\dfrac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ が直交していることから分かる

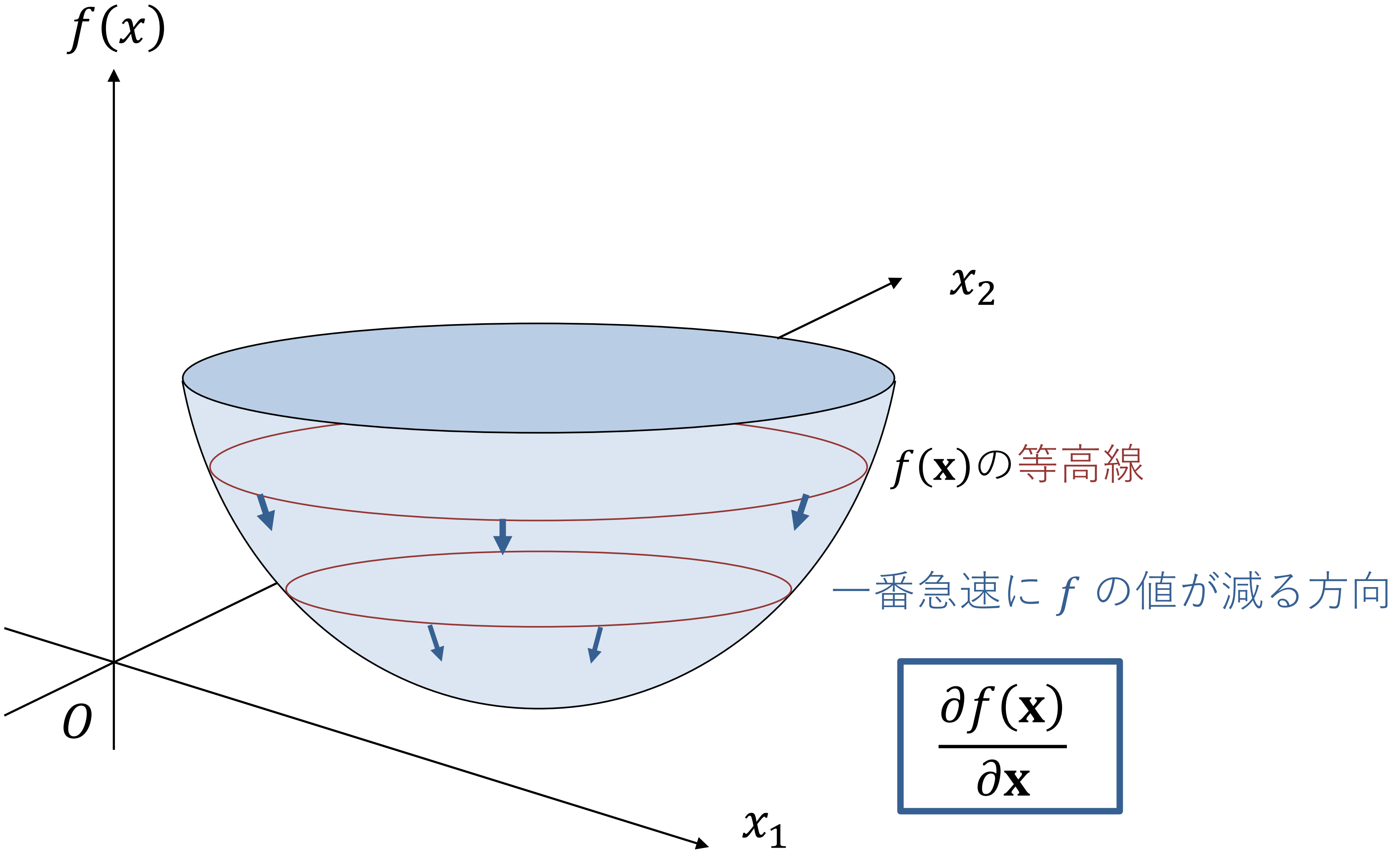
その向きに $\mathbf{x}$ を少し変えても f の値は変化しないため、

$$\delta f = \delta\mathbf{x} \cdot \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = 0 \quad \Leftrightarrow \quad \delta\mathbf{x} \text{ と } \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \text{ は直交}$$

$-\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}$が最急降下方向であること(2/2)



復習：
最急降下法のイメージ



まとめ

- 線形分離不可能な問題について考えた。
- 線形ニューロン二層ネットワークにおける学習法であるデルタ則について説明した。
- デルタ則が何をやっているか数学的に理解した。

復習問題

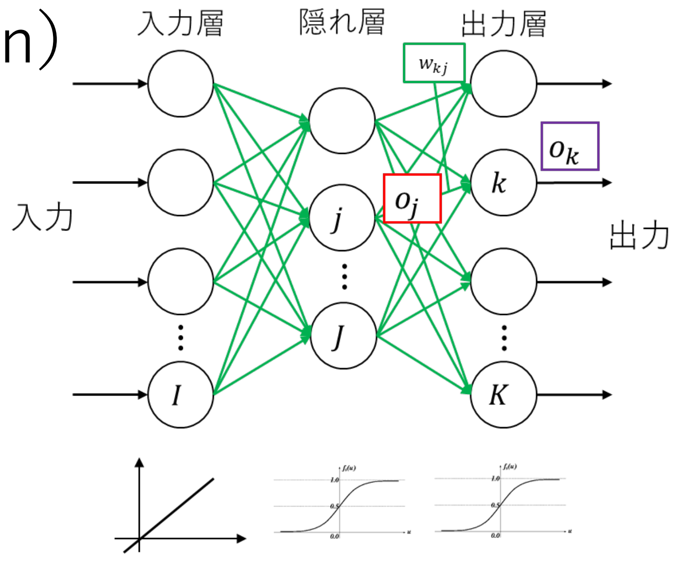
1. 論理関数XORは線形分離可能か？

2. 連続値をとるA層とR層の2層からなるネットワークに誤り訂正学習法を拡張して適用したアルゴリズムは何か？

3. 上記のアルゴリズムは最適化問題における何という手法と解釈する事ができるか？また、それはなぜか？

次回の講義

- 多層ニューラルネットワーク
  - シグモイド関数の三層ニューラルネットワーク
  - 一般化デルタ則
  - 誤差逆伝搬法（Back propagation）



f：線形関数　f：シグモイド関数

$$o_k = f(u_k)$$

$$u_k = \sum_{j=1}^{J} w_{kj} o_j$$

荷重和＋非線形変換

学習で可変な結合荷重