Section 3

- サポートベクトルマシン (6章)
- ニューラルネットワーク (7章)

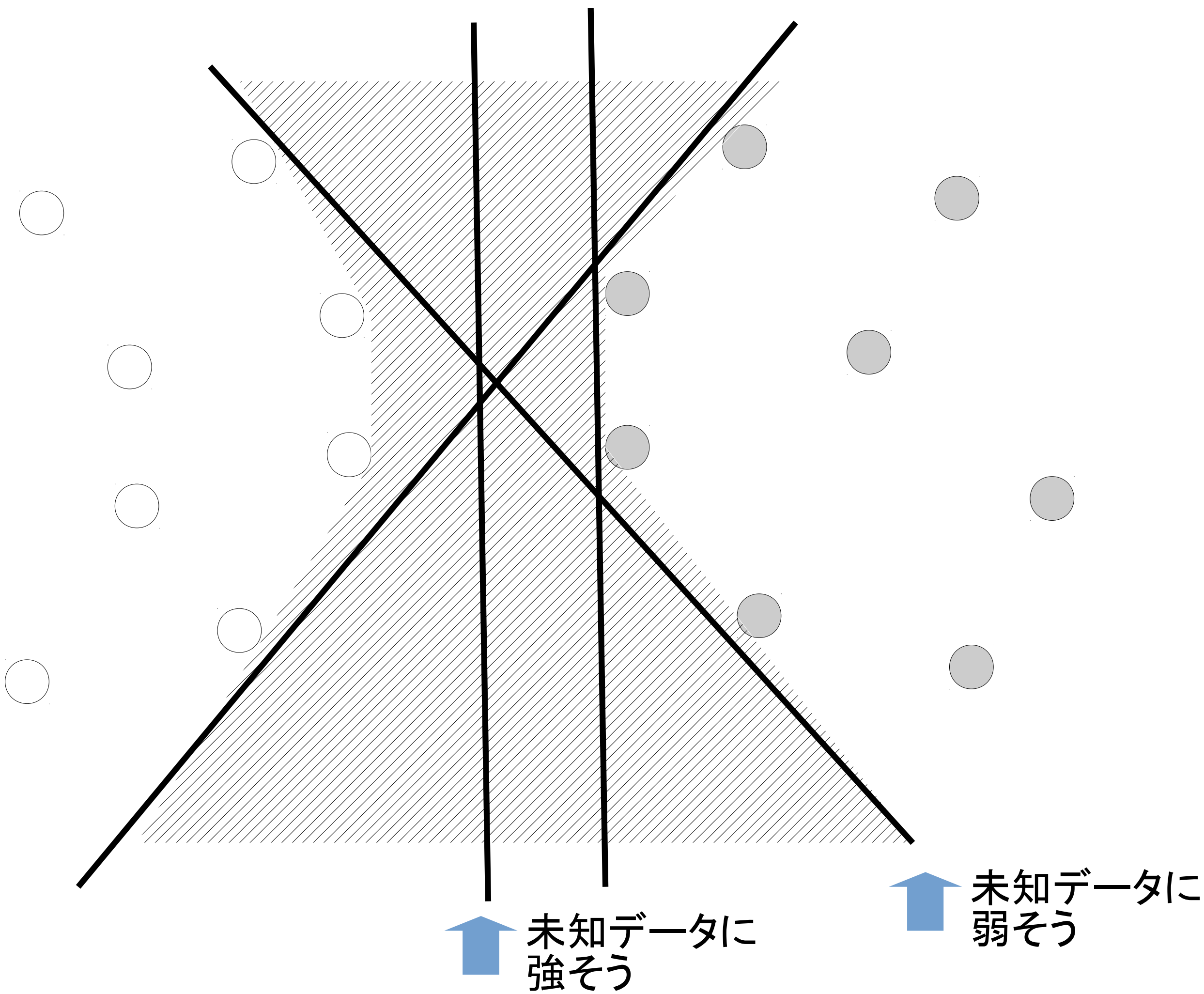
6. 限界は破れるか（１）
－ サポートベクトルマシン －

- パーセプトロンの学習規則の限界
  - 学習パターンが線形分離可能である場合は識別面が見つかるが、信頼できる識別面とは限らない
  - 学習パターンが線形分離不可能である場合は、学習が停止しない

➡ サポートベクトルマシン（SVM）

6.1　識別面は見つかったけれど

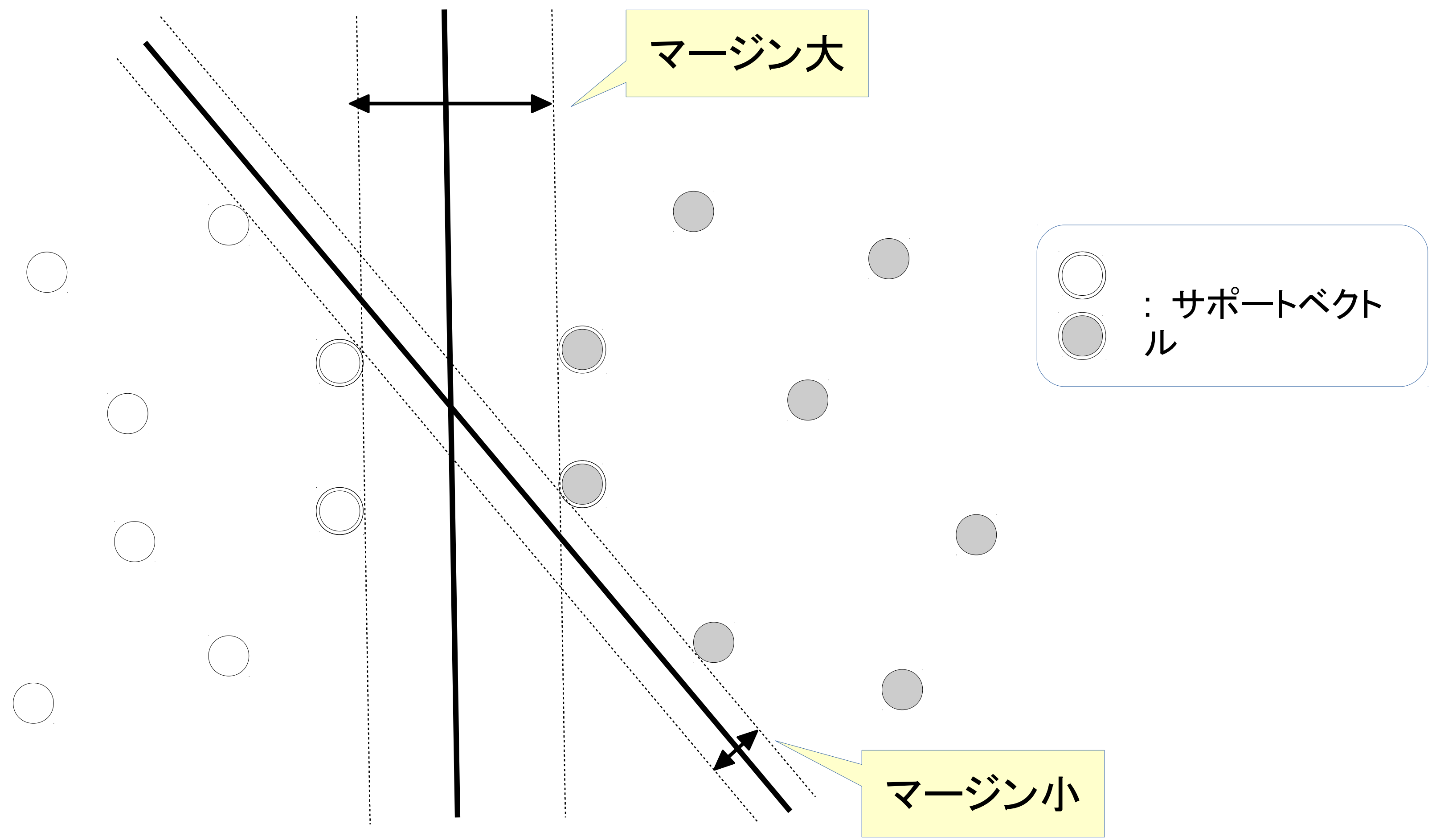
パーセプトロンの学習規則ではどれが見つかるかわからない



6.2　サポートベクトルマシンの学習アルゴリズム

6.2.1 サポートベクトル

- 線形 SVM
  - マージン最大となる線形識別面を求める



6.2.2　マージンを最大にする

- 学習データ

$$\{(\boldsymbol{x}_i, y_i)\} \quad i = 1, \ldots, n, \quad y_i = 1 \ or \ -1$$

- 線形識別面の式

$$\boldsymbol{w}^T \boldsymbol{x} + w_0 = 0$$

- 識別面の制約の導入（係数を定数倍しても平面は不変）

$$\min_{i=1,\ldots,n} |\boldsymbol{w}^T \boldsymbol{x}_i + w_0| = 1$$

- 学習パターンと識別面との最小距離（＝マージン）

$$\min_{i=1,\ldots,n} Dist(\boldsymbol{x}_i) = \min_{i=1,\ldots,n} \frac{|\boldsymbol{w}^T \boldsymbol{x}_i + w_0|}{||\boldsymbol{w}||} = \frac{1}{||\boldsymbol{w}||}$$

これを
最大化

点と直線の距離の公式

$$r = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}}$$

6.2.2　マージンを最大にする

- 目的関数の置き換え：$\min \frac{1}{2}||\boldsymbol{w}||^2$

極値が求めやすい２乗に

- 制約条件：　$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 \quad i = 1, \ldots, n$
- 解法：ラグランジュの未定乗数法
  - 問題　$\min f(x) \quad s.t.\ g(x) = 0$
  - ラグランジュ関数　$L(x, \alpha) = f(x) - \alpha g(x)$
    - $\alpha \geq 0$
    - x,α で偏微分して０になる値が極値

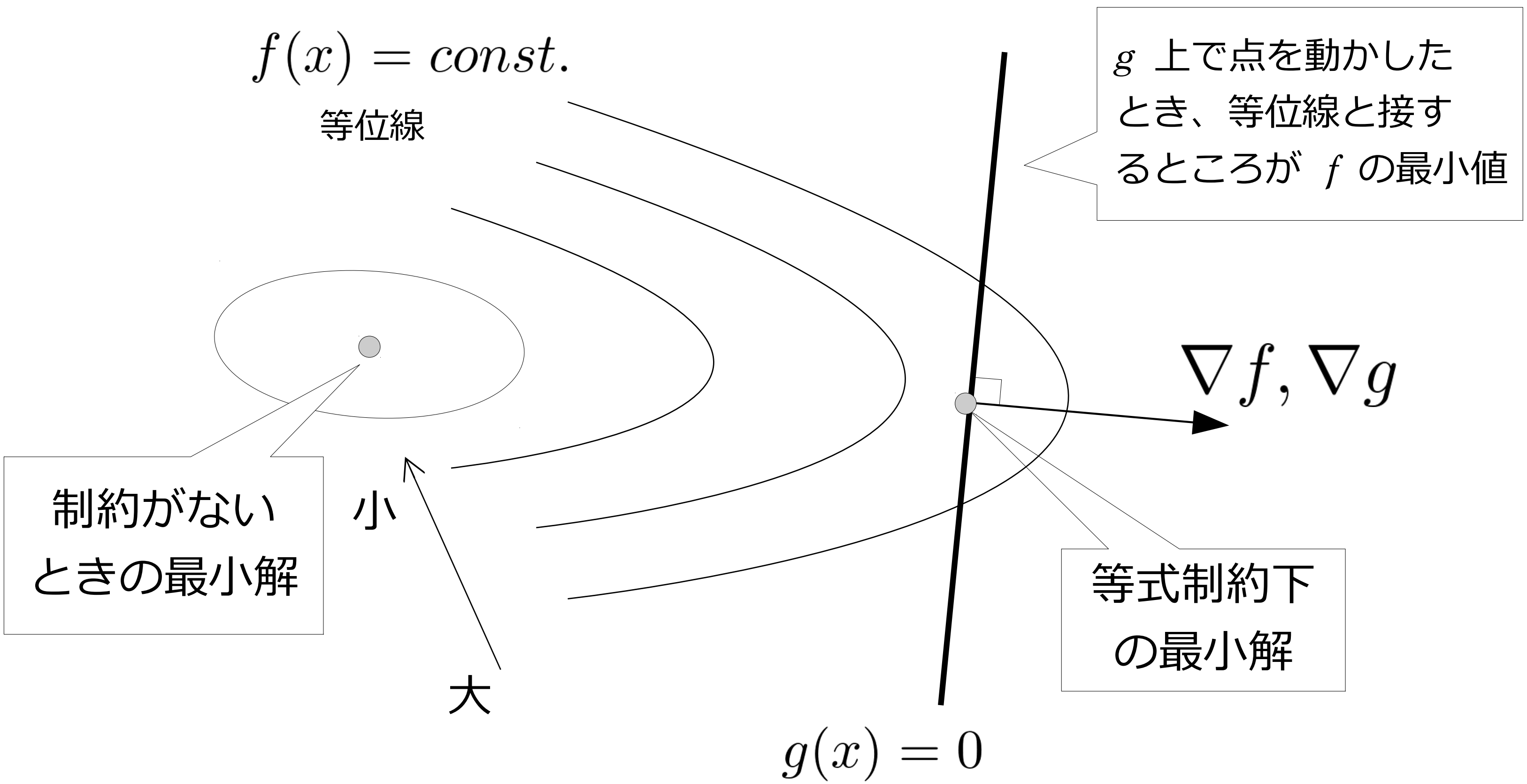
ラグランジュの未定乗数法（付録 A.4 ）

$$\min f(x) \quad s.t.\ g(x) = 0$$

$$L(x, \alpha) = f(x) - \alpha g(x)$$

$$\frac{\partial L(\boldsymbol{x}, \alpha)}{\partial \boldsymbol{x}} = \nabla f(\boldsymbol{x}) - \alpha \nabla g(\boldsymbol{x}) = 0$$

$$\frac{\partial L(\boldsymbol{x}, \alpha)}{\partial \alpha} = -g(\boldsymbol{x}) = 0$$



6.2.2　マージンを最大にする

- 計算

$$L(\boldsymbol{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{n} \alpha_i (y_i(\boldsymbol{w}^T \boldsymbol{x} + w_0) - 1)$$

$$\frac{\partial L}{\partial w_0} = 0 \quad \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \quad \Rightarrow \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$$

$$L(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j - \sum_{i=1}^{n} \alpha_i$$

$$\alpha_i \geq 0$$

最大化 2 次計画問題
→ $\boldsymbol{\alpha}$ が求まる

6.2.2　マージンを最大にする

- 定数項の計算
  - 各クラスのサポートベクトルから求める

$$w_0 = -\frac{1}{2}(\boldsymbol{w}^T \boldsymbol{x}_{s1} + \boldsymbol{w}^T \boldsymbol{x}_{s2})$$

- 識別関数

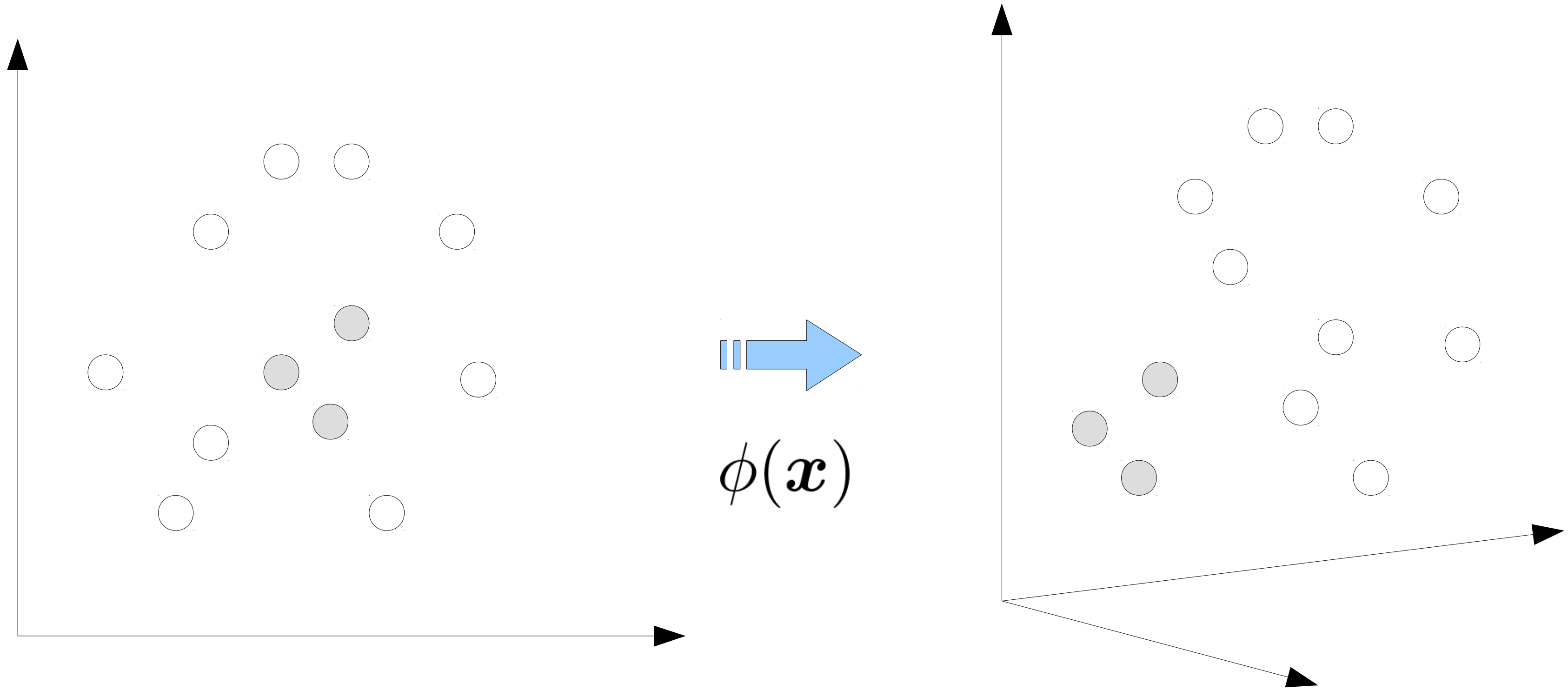
$$\begin{aligned} g(\boldsymbol{x}) &= \boldsymbol{w}^T \boldsymbol{x} + w_0 \\ &= \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}^T \boldsymbol{x}_i + w_0 \end{aligned}$$

サポートベクトルに対応する α_i のみが 0 以上、残りは 0

6.3　線形分離可能にしてしまう

6.3.1　高次元空間への写像

- 特徴ベクトルの次元数を増やす



2 次元で線形分離不可能なデータ

線形分離可能な 3 次元へ

ただし、元の空間でのデータ間の
距離関係は保持するように

6.3.2　カーネル法

- 非線形変換関数： $\phi(\boldsymbol{x})$
- カーネル関数（類似度関数のようなもの）
  - 元の空間での距離が変換後の空間の内積に対応

$$K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}')$$

線形カーネル
$(\boldsymbol{x}^T \boldsymbol{x}')^p$
を用いる場合もある

  - カーネル関数の例
    - 多項式カーネル $K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}' + 1)^p$
    - ガウシアンカーネル $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{\sigma^2})$

これらの形であれば、対応する非線形変換が
存在することが数学的に保証されている

6.3.2　カーネル法

- 変換後の識別関数：　$g(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + w_0$
- SVM で求めた　$\boldsymbol{w}$　の値を代入

$$
\begin{aligned}
g(\boldsymbol{x}) &= \sum_{i=1}^{n} \alpha_i y_i \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}_i) + w_0 \\
&= \sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + w_0
\end{aligned}
$$

非線形変換の
式は不要！！！

カーネルトリック

6.3.3　具体的なカーネル関数

- 線形カーネル（２次）の展開

$$
\begin{aligned}
K(\boldsymbol{x}, \boldsymbol{x}') &= (\boldsymbol{x}^T \boldsymbol{x}')^2 \\
&= (x_1 x_1' + x_2 x_2')^2 \\
&= x_1^2 {x_1'}^2 + 2x_1 x_2 x_1' x_2' + {x_2}^2 {x_2'}^2 \\
&= (x_1^2, \sqrt{2} x_1 x_2, x_2^2) \cdot ({x_1'}^2, \sqrt{2} x_1' x_2', {x_2'}^2)
\end{aligned}
$$

- 多項式カーネル（２次）の展開

$$
\begin{aligned}
K(\boldsymbol{x}, \boldsymbol{x}') &= (\boldsymbol{x}^T \boldsymbol{x}' + 1)^2 \\
&= (x_1 x_1' + x_2 x_2 + 1')^2 \\
&= x_1^2 {x_1'}^2 + {x_2}^2 {x_2'}^2 + 2x_1 x_2 x_1' x_2' + 2x_1 x_1' + \ 2x_2 x_2' + 1 \\
&= (x_1^2, x_2^2, \sqrt{2} x_1 x_2, \sqrt{2} x_1, \sqrt{2} x_2, 1) \cdot \\
&\quad\ (x_1'^2, x_2'^2, \sqrt{2} x_1' x_2', \sqrt{2} x_1', \sqrt{2} x_2', 1)
\end{aligned}
$$

7．限界は破れるか（２）
－　ニューラルネットワーク　－

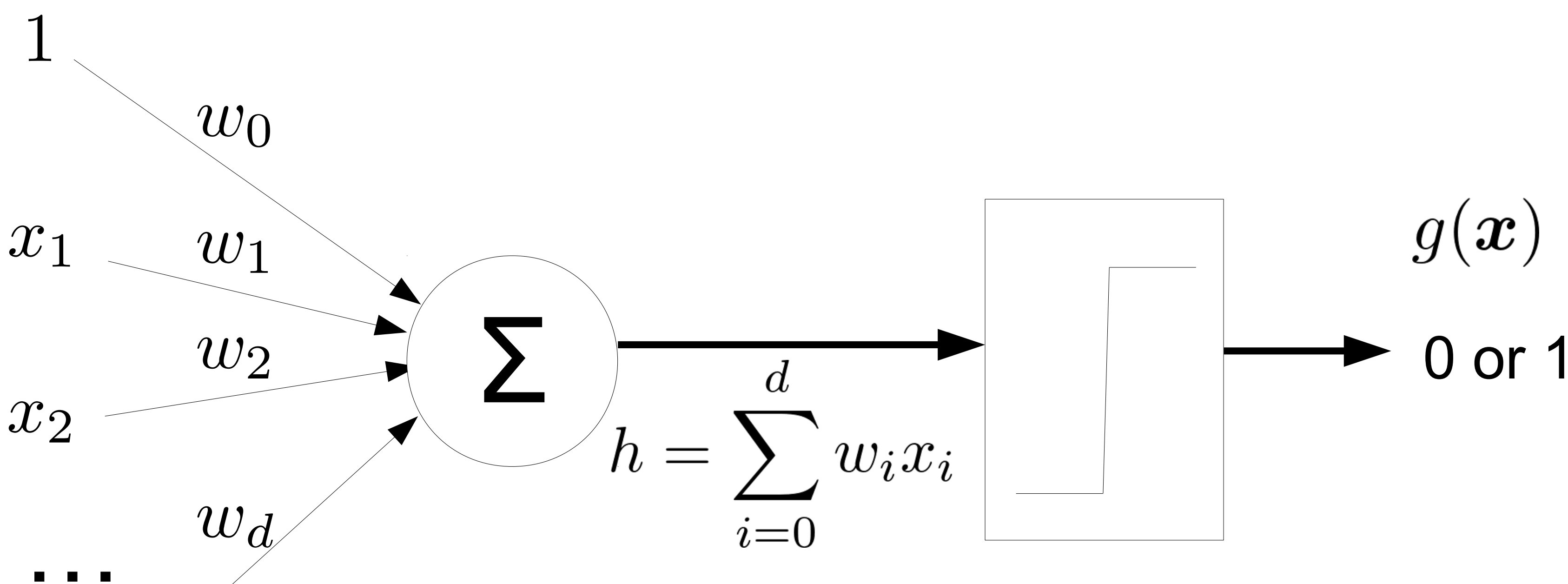
- 誤差評価に基づく学習
  - 誤差最小・任意形の識別面を学習することはできないか ➡ ニューラルネットワーク

7.1　ニューラルネットワークの構成

- 単層パーセプトロンの定義

以後、$\boldsymbol{w}$ は w_0 を含む

  - $\boldsymbol{w}^T\boldsymbol{x} = 0$　という特徴空間上の識別面を表現



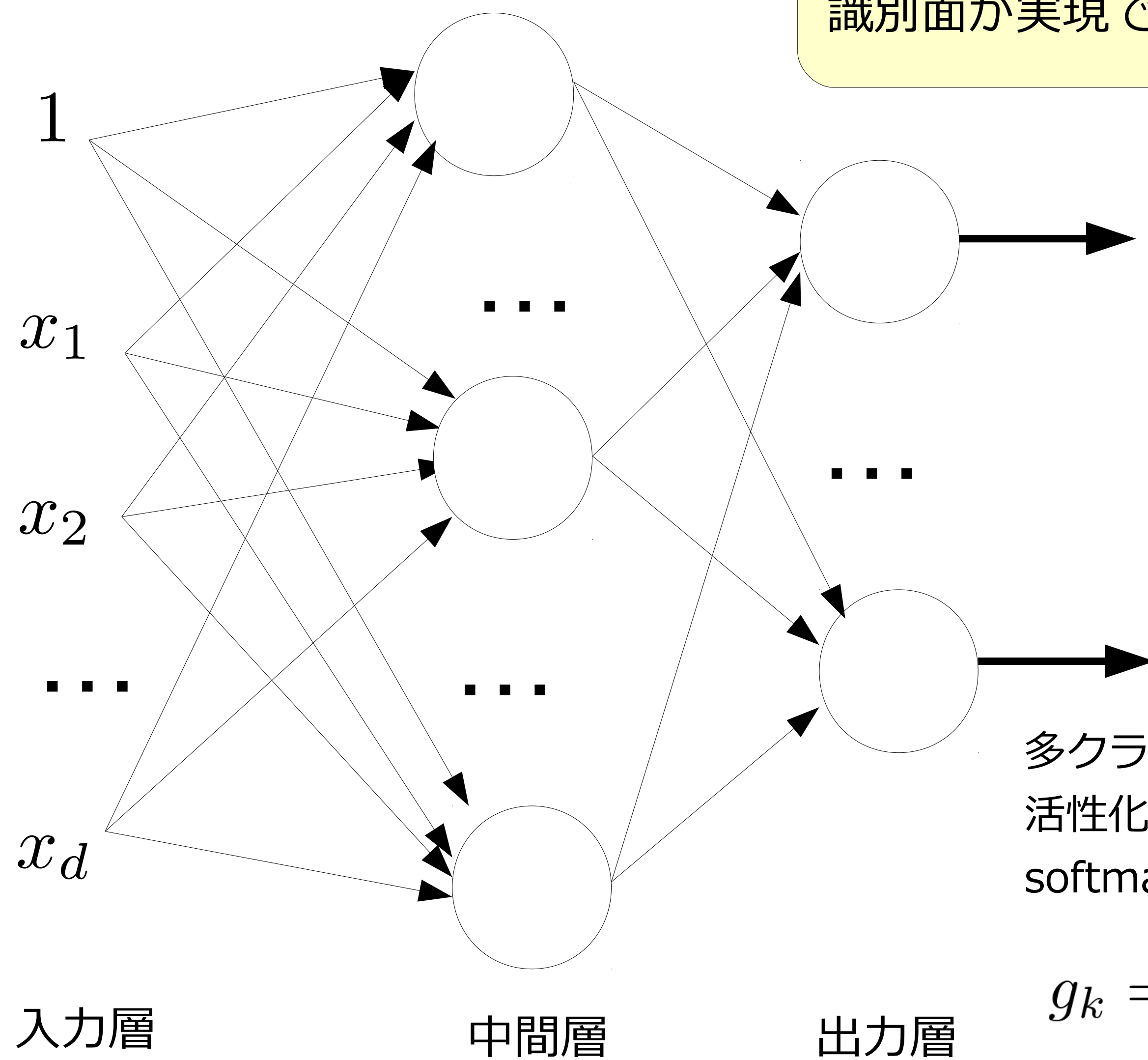
閾値関数

$$f(h) = \begin{cases} 0 & (h < 0) \\ 1 & (h \geq 0) \end{cases}$$

7.1 ニューラルネットワークの構成

- 多層パーセプトロン

特徴空間上で複雑な非線形識別面が実現できる

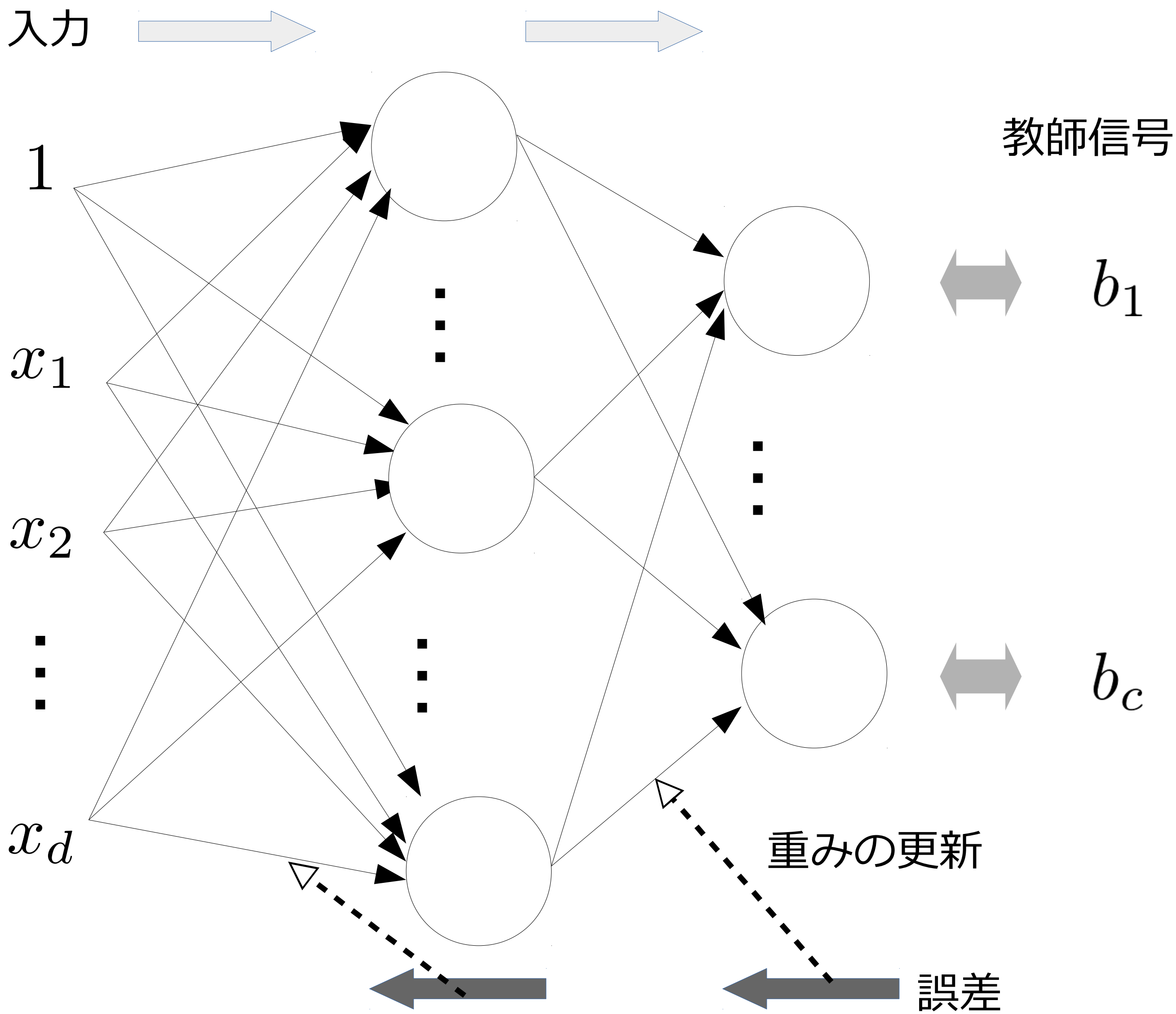


多クラス問題の出力層には活性化関数として以下の softmax 関数を用いる

$$g_k = \frac{\exp(h_k)}{\sum_{j=1}^{c} \exp(h_j)}$$

7.2 誤差逆伝播法による学習

- 誤差逆伝播法の名前の由来



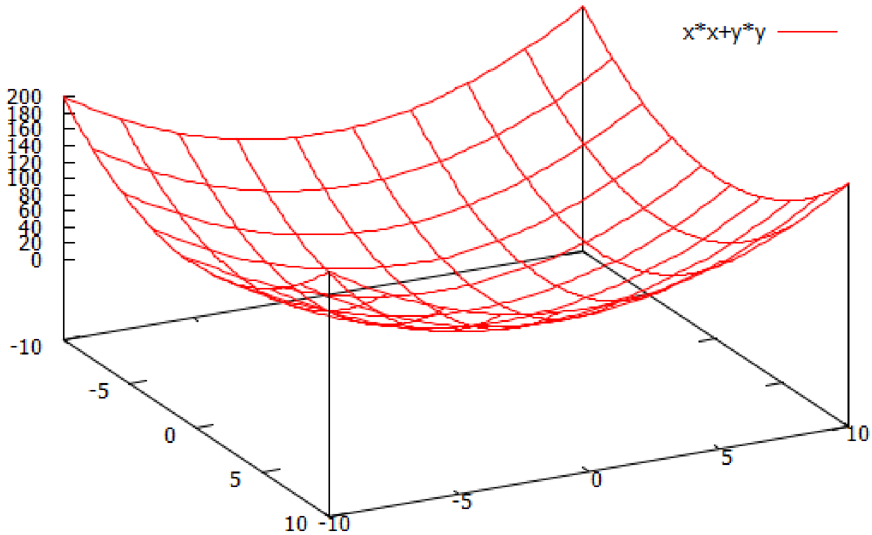
7.2 誤差逆伝播法による学習

- 結合重みの調整アルゴリズム
  - 二乗誤差

$$J(\boldsymbol{w}) \equiv \frac{1}{2}\sum_{p=1}^{n}(g(\boldsymbol{x}_p) - b_p)^2$$

全データに対する正解と関数の出力との差の２乗和

  - $\boldsymbol{J}$ は $\boldsymbol{w}$ の関数
    - $\boldsymbol{w}$ を J の勾配方向へ一定量だけ動かすことを繰り返して、最適解へ収束させる（→最急降下法）



$$\boldsymbol{w}' \leftarrow \boldsymbol{w} - \rho\frac{\partial J}{\partial \boldsymbol{w}}$$

ただし、ニューラルネットワークによる識別面は非線形なので、誤差関数はもっと複雑な形

7.2 誤差逆伝播法による学習

1. リンクの重みを小さな初期値に設定
2. 個々の学習データ $(\boldsymbol{x}_p, b_p)$に対して以下繰り返し

   a)入力 $\boldsymbol{x}_p$ に対するネットワークの出力 $\boldsymbol{g}_p$ を計算

   b)出力層の k 番目のユニットに対してエラー量 ε を計算

$$\varepsilon_k \leftarrow (g_k - b_k)g_k(1 - g_k)$$

   c)中間層の h 番目のユニットに対してエラー量 ε を計算

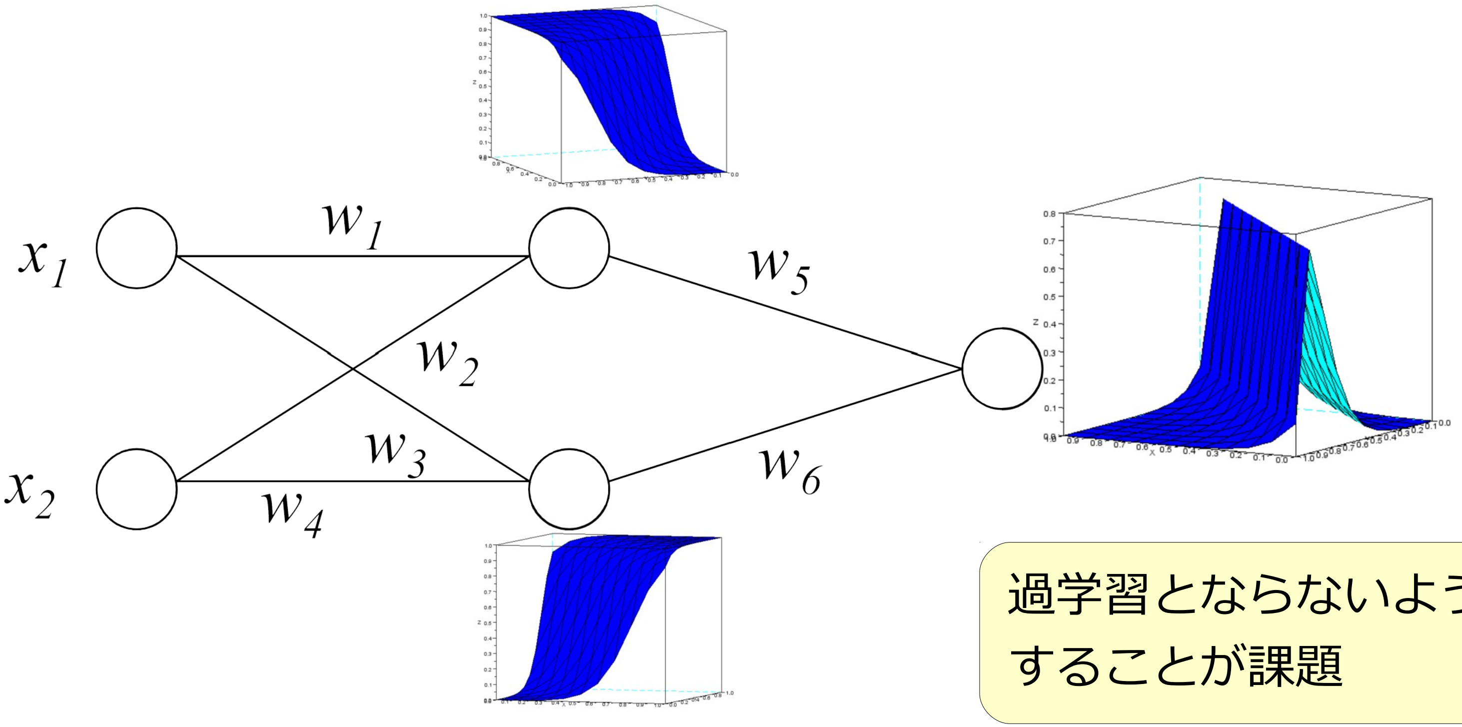
$$\varepsilon_j \leftarrow (\sum_k \varepsilon_k w_k)g_j(1 - g_j)$$

   d)重みの更新

$$w_{ji} \leftarrow w_{ji} + \rho\varepsilon_j x_{pj}$$

7.2 誤差逆伝播法による学習

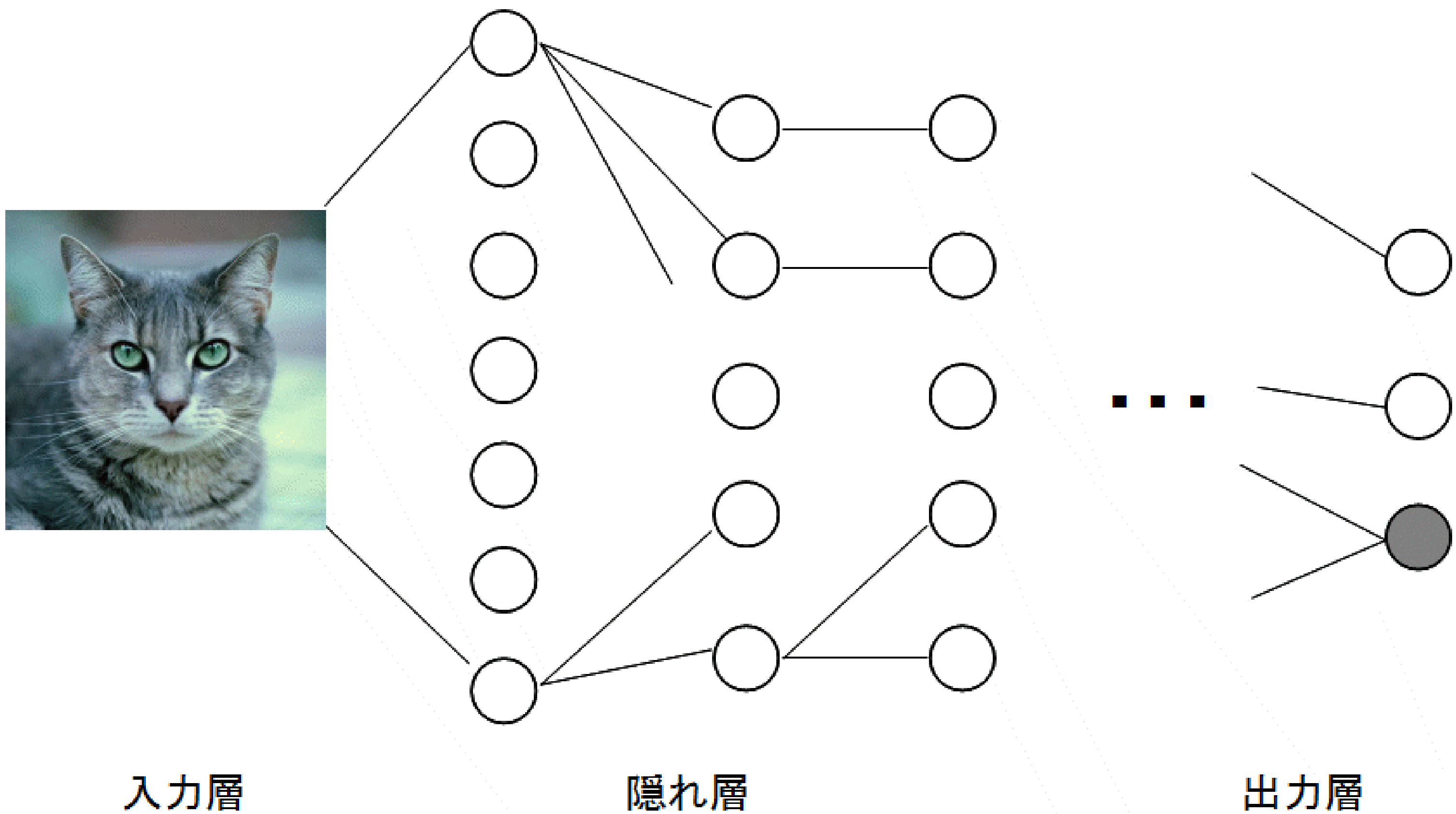
- 識別面の複雑さ
  - 中間層のユニット数に関係する
  - シグモイド関数（非線形）を任意の重み・方向で足し合わせることで複雑な非線形識別面を構成



過学習とならないようにすることが課題

7.3 ディープニューラルネットワーク

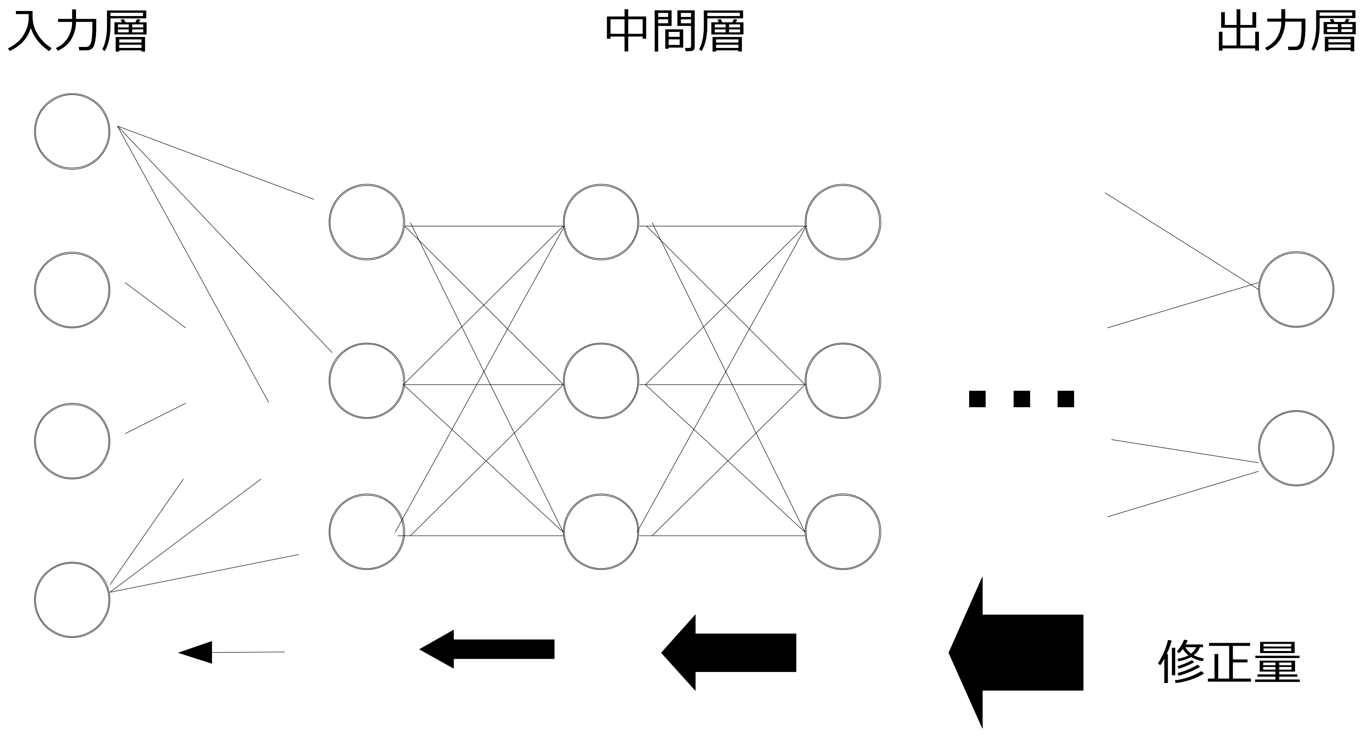
- 深層学習：多階層ニューラルネットによる学習
  - 表現学習：抽出する特徴も学習する



7.3.1 勾配消失問題とは

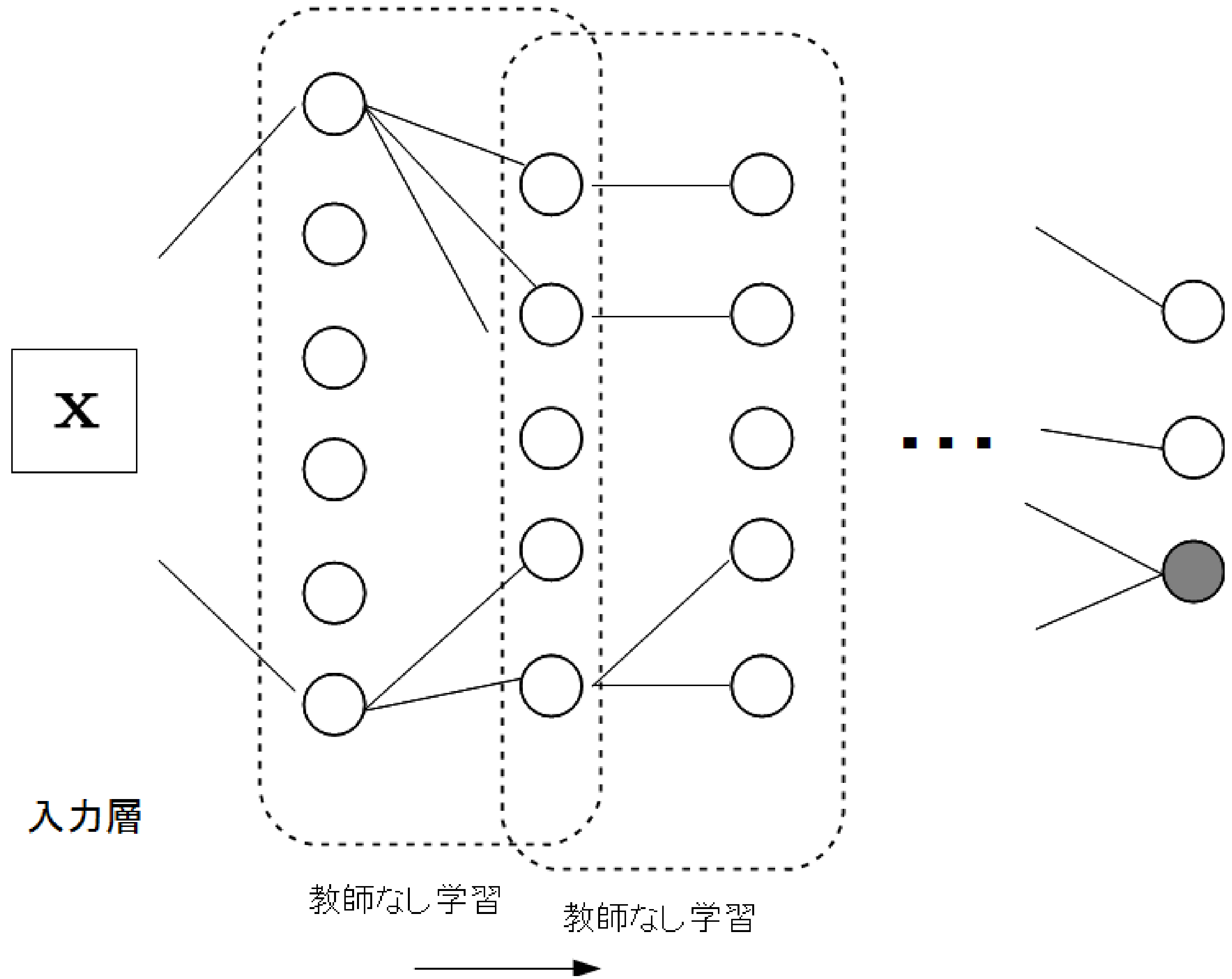
- 多階層における誤差逆伝播法の問題点
  - 修正量が消失／発散する

順方向：非線形　　逆方向：線形



7.3.2 多階層学習における工夫

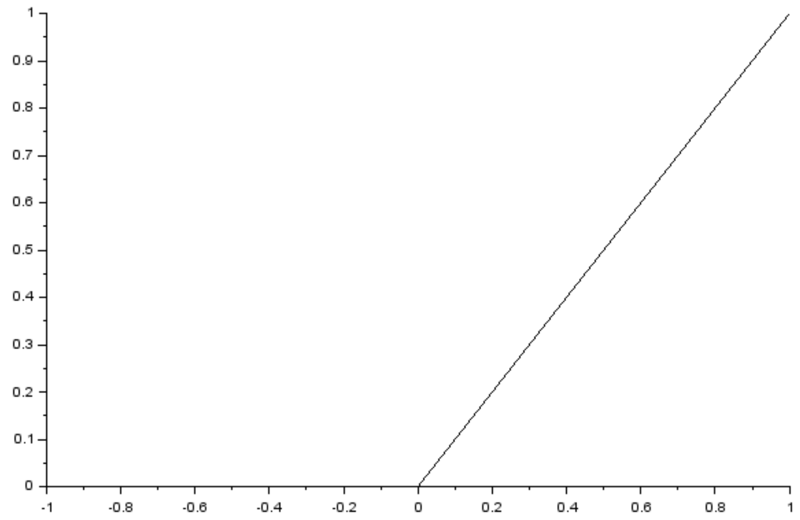
- 事前学習法
  - 深層学習における初期パラメータ学習



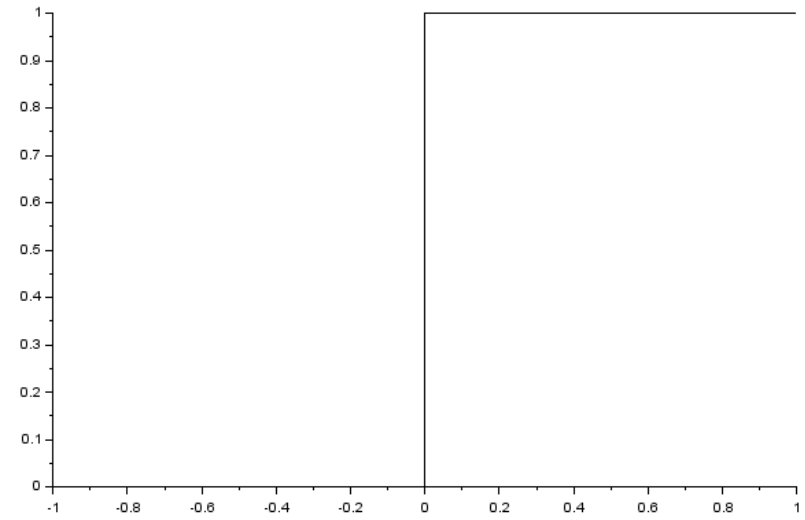
7.3.2 多階層学習における工夫

- 活性化関数を rectified linear 関数に ➡ RELU

$$f(x) = \max(0, x)$$



(a) rectified linear 関数

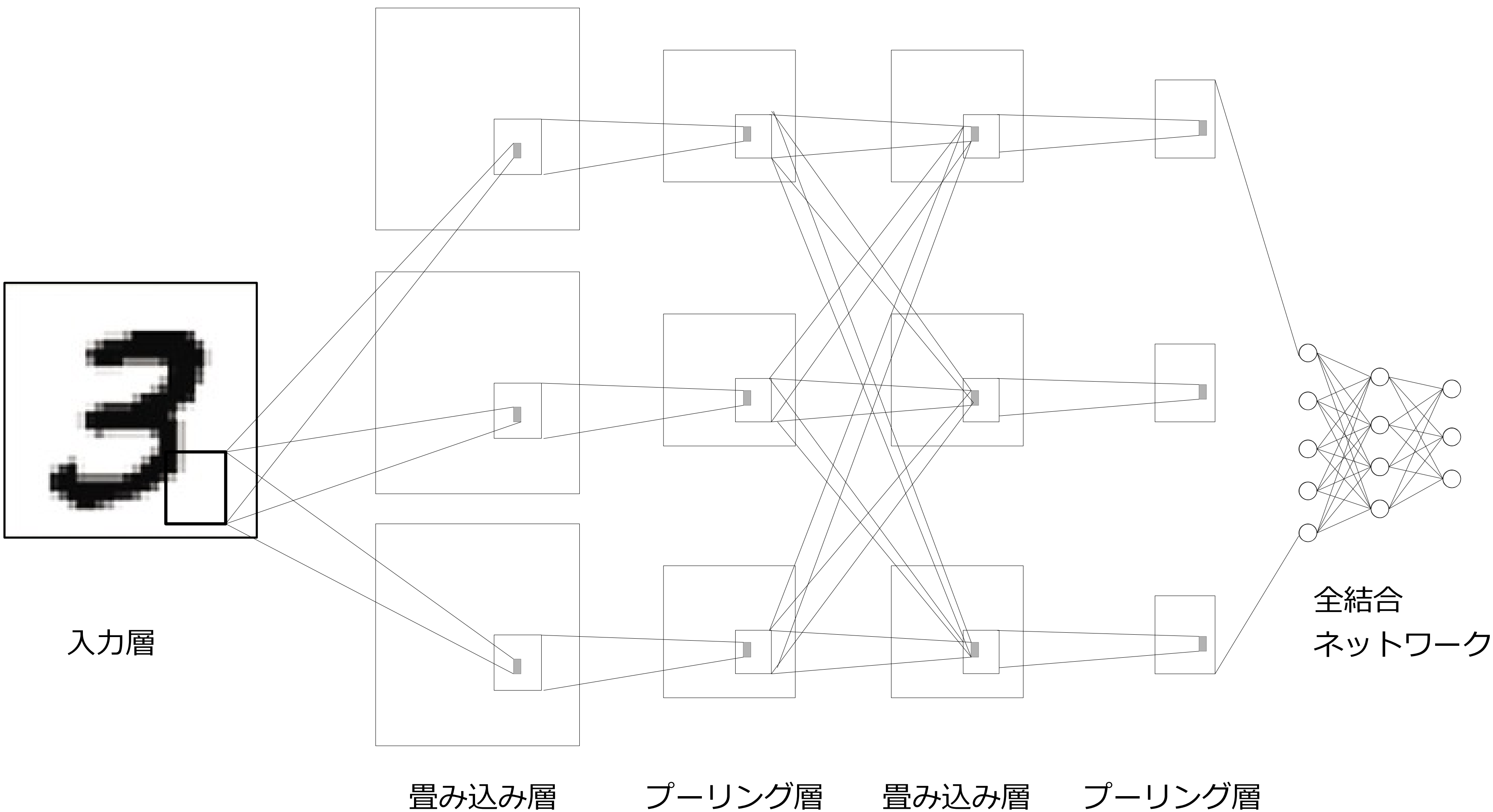


(b) (a) の導関数

- RELU の利点
  - 誤差消失が起こりにくい
  - 0 を出力するユニットが多くなる

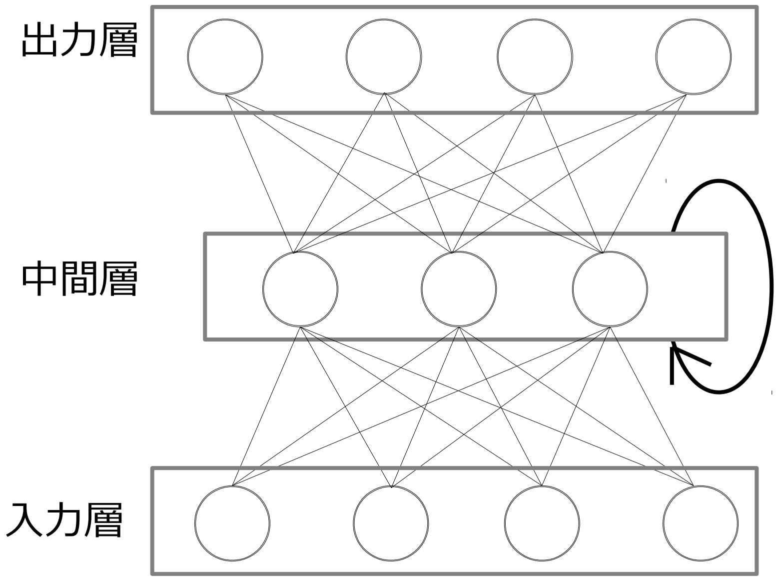
7.3.3 特化した構造をもつニューラルネットワーク

- 畳み込みニューラルネットワーク
  - 画像認識に適する

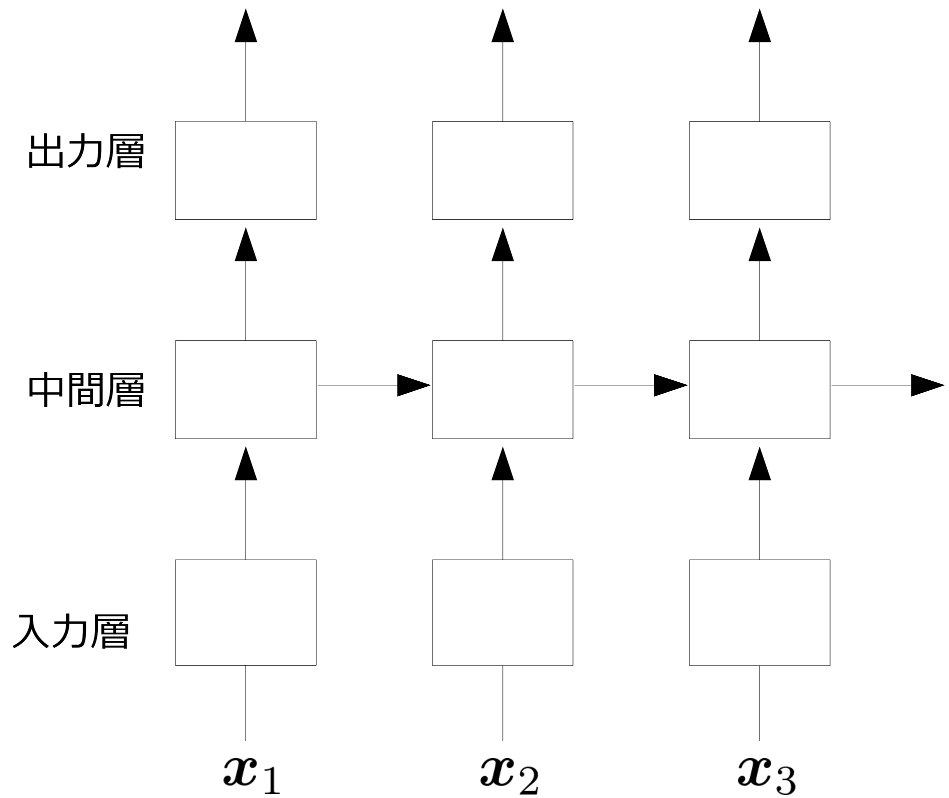


7.3.3 特化した構造をもつニューラルネットワーク

- リカレントニューラルネットワーク
  - 時系列信号の認識や自然言語処理に適する



(a) リカレントニューラルネットワーク



(b) 帰還路を時間方向に展開

Section3 のまとめ

- サポートベクトルマシン
  - 低次元で識別しにくいデータを、高次元の疎らなデータに変換し、マージン最大の線形識別面を求める手法
  - 高次元特徴に強く、大局的最適解が求まる
- ニューラルネットワーク
  - 複雑な非線形識別面を求める
  - 局所最適解で学習が終了したり、過学習に陥りやすいという欠点があった
  - 現在は、ディープニューラルネットワークの発展によりパターン認識の中心技術になった