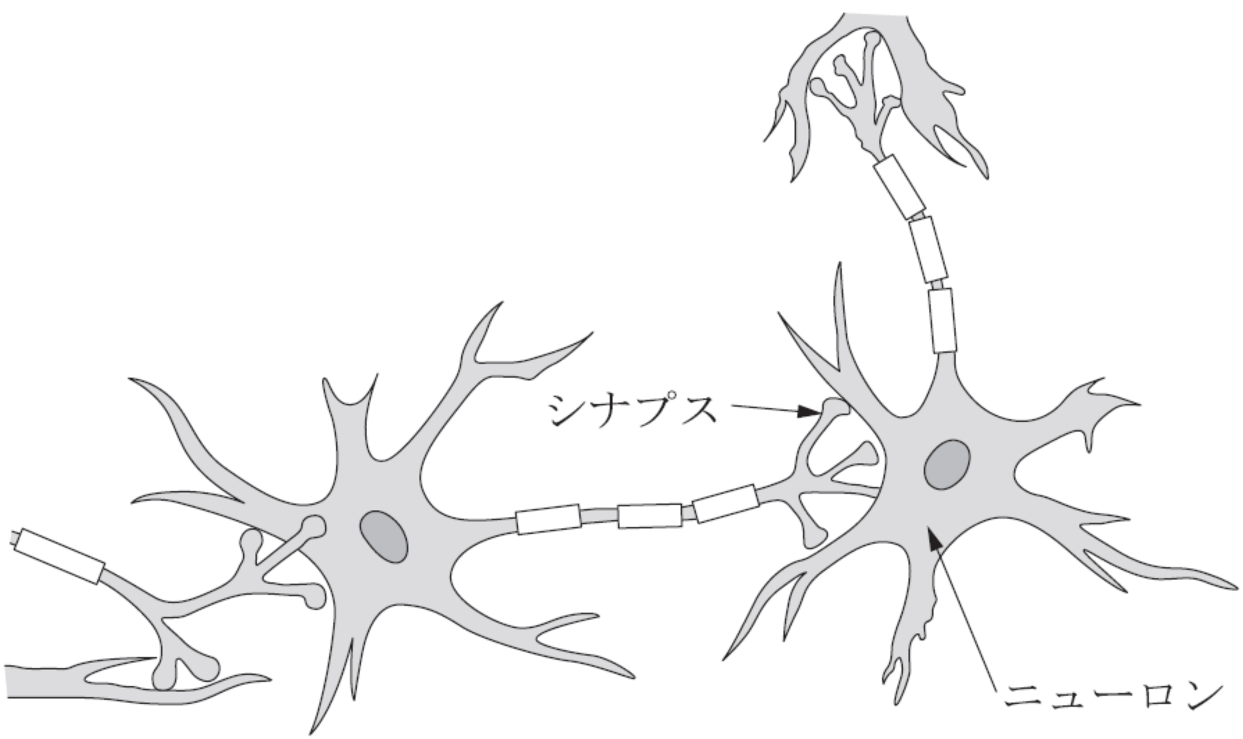
7. 限界は破れるか（2）
－ ニューラルネットワーク －

- 誤差評価に基づく学習
  - 誤差最小の線形識別面を学習できる
  - 非線形識別面の学習は可能だが、事前にどのような非線形関数にするかを設計する必要がある
  - 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

- 神経細胞の計算メカニズム
  - ニューロンがシナプス結合によって複雑に結合
  - 入力された電気信号の重み付き和の値によって、各ニューロンの活性／非活性が決まる

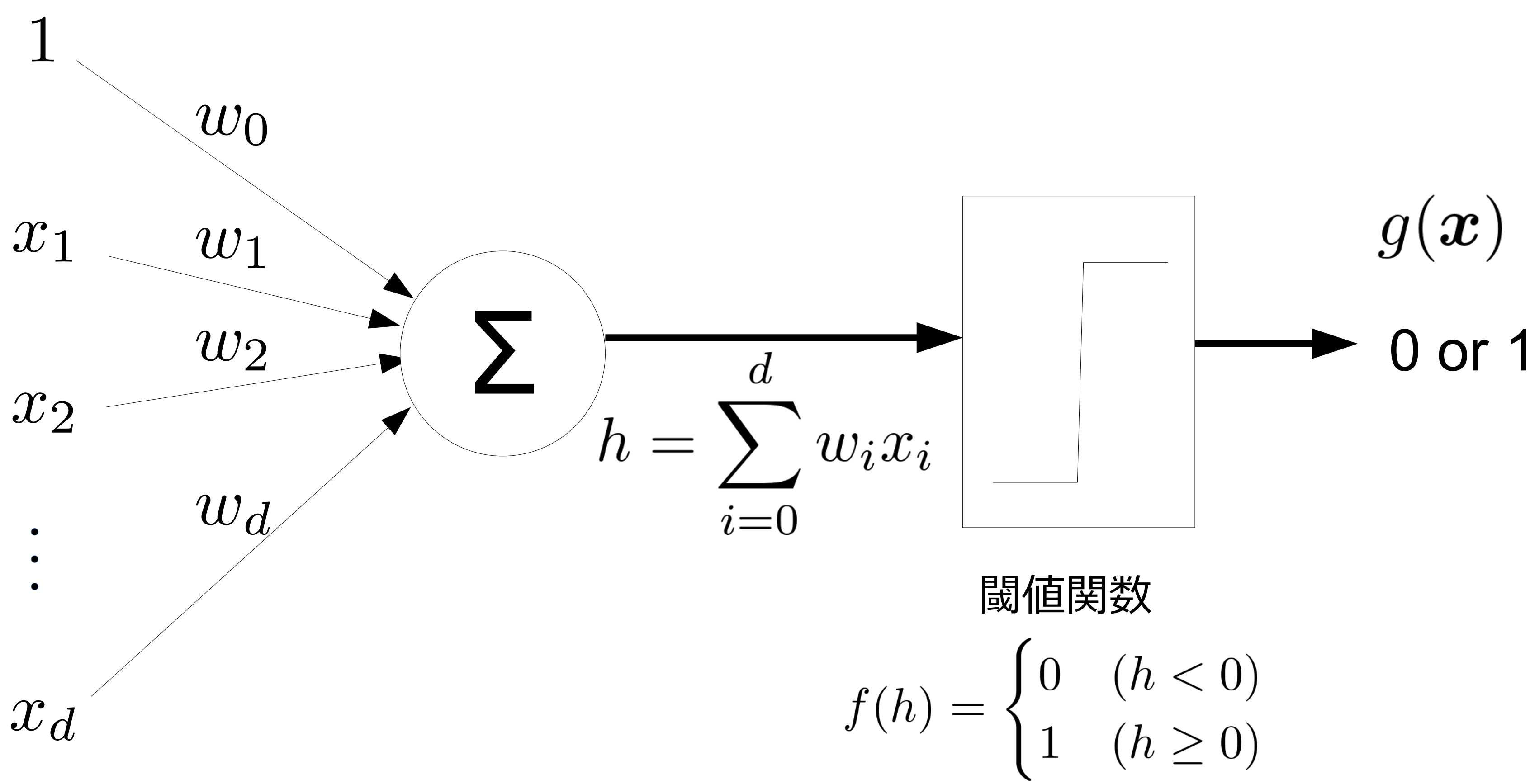


7.1 ニューラルネットワークの構成

- 閾値論理ユニットによるニューロンのモデル化
  - $\boldsymbol{w}^T\boldsymbol{x} = 0$ という特徴空間上の識別面を表現

    → パーセプトロン

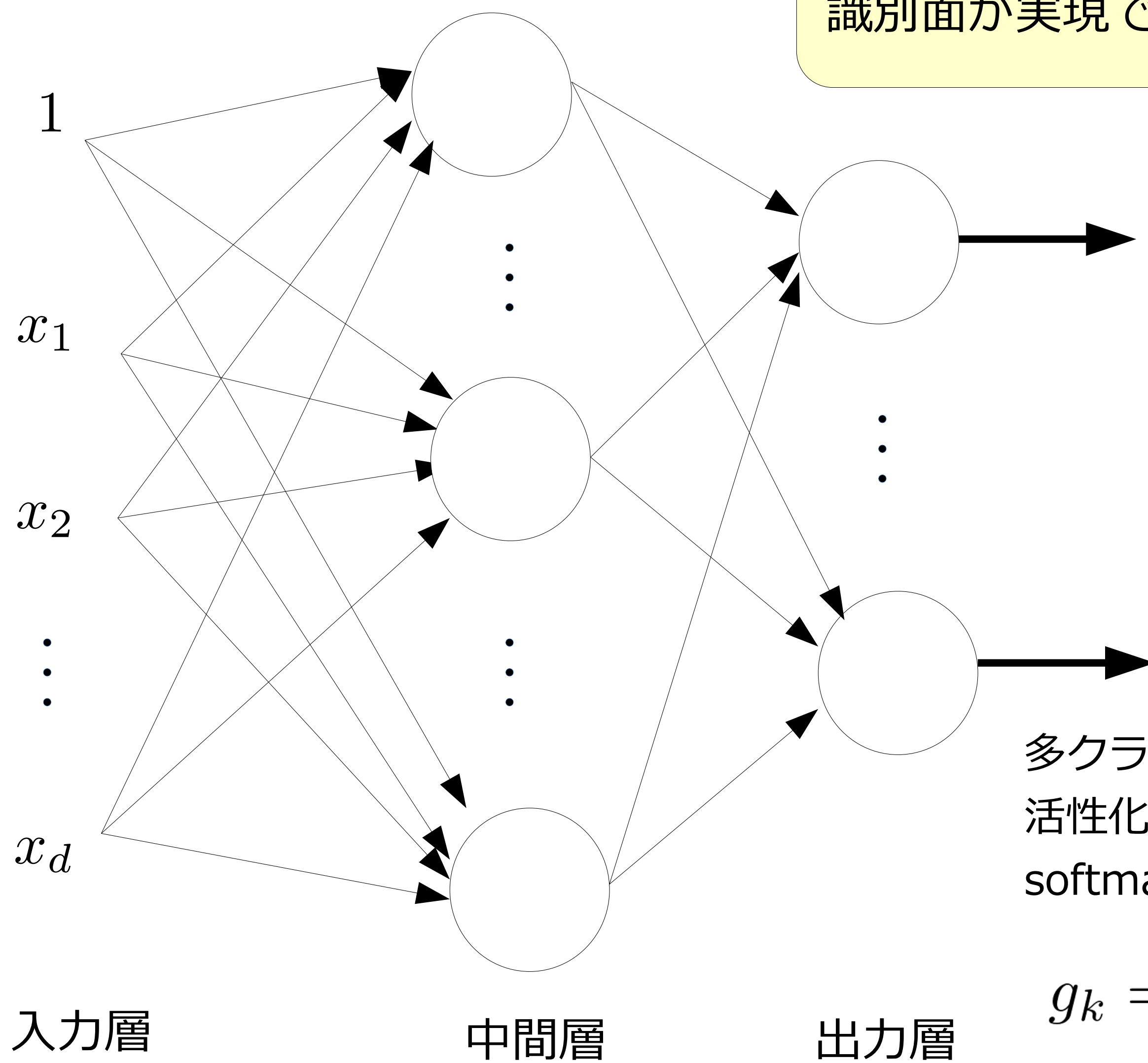
$\boldsymbol{w}$ は w_0 を含む



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

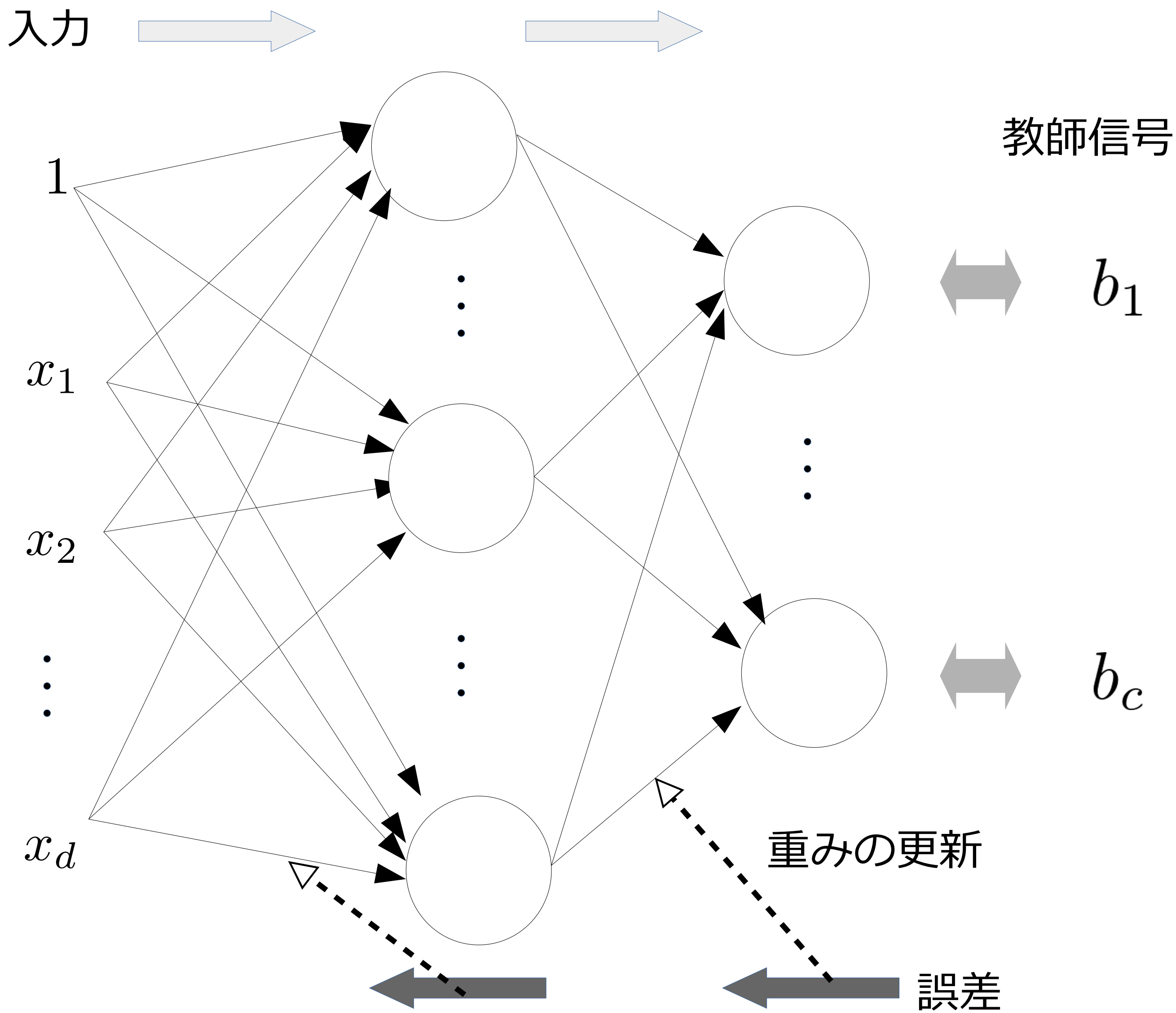


多クラス問題の出力層には活性化関数として以下のsoftmax関数を用いる

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

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



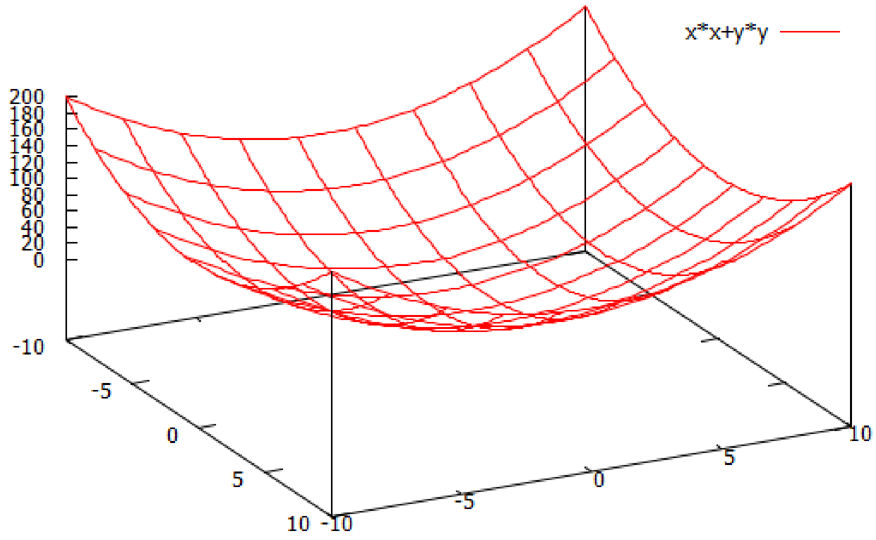
7.2 誤差逆伝播法による学習

- 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）

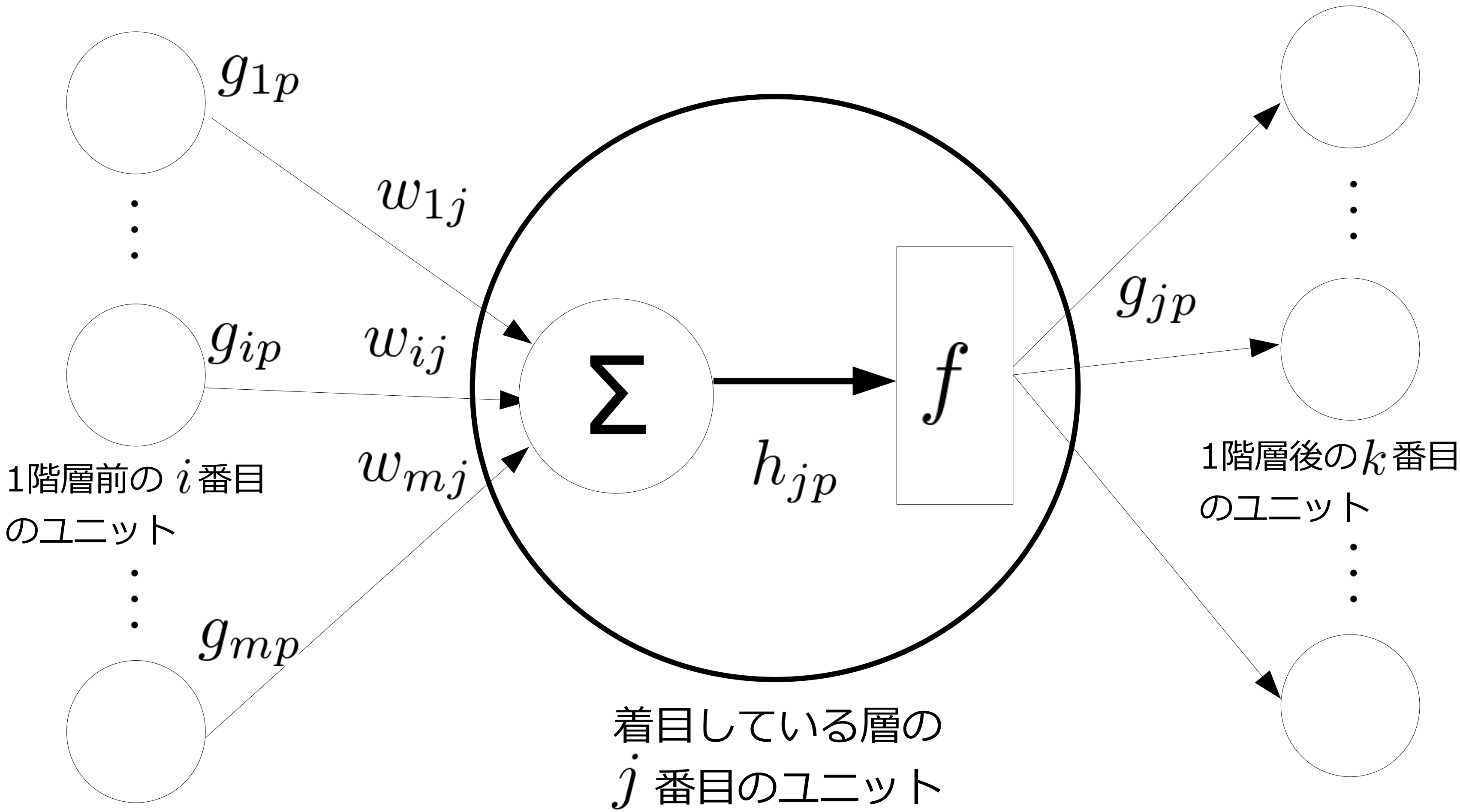


$$\boldsymbol{w}' \leftarrow \boldsymbol{w} - \rho\frac{\partial J}{\partial \boldsymbol{w}}$$

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

- 閾値論理ユニットの入出力



7.2 誤差逆伝播法による学習

- 学習パターン $\boldsymbol{x}_p$ が入力されたときのユニット j の入力

$$h_{jp} = \sum_i w_{ij} g_{ip}$$

- ユニット j の出力

$$g_{jp} = f(h_{jp})$$

- 出力層における誤差の定義

$$J_p = \frac{1}{2} \sum_l (g_{lp} - b_{lp})^2$$

- ユニット j の重みの調整式

$$w'_{ij} = w_{ij} - \rho \frac{\partial J_p}{\partial w_{ij}}$$

ユニット j の重み
が変化すれば、
誤差も変化する

7.2 誤差逆伝播法による学習

- 調整量の計算

$$\frac{\partial J_p}{\partial w_{ij}} = \frac{\partial J_p}{\partial h_{jp}} \cdot \frac{\partial h_{jp}}{\partial w_{ij}}$$

g_{ip}

合成関数 $f(g(x))$ の微分

$y = f(u), u = g(x)$

と分けて

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$$

- 右辺第1項を ε_{jp} とおく

$$\varepsilon_{jp} = \frac{\partial J_p}{\partial h_{jp}} = \frac{\partial J_p}{\partial g_{jp}} \cdot \frac{\partial g_{jp}}{\partial h_{jp}} = \frac{\partial J_p}{\partial g_{jp}} \cdot f'(h_{jp})$$

- ユニット j が出力層の場合

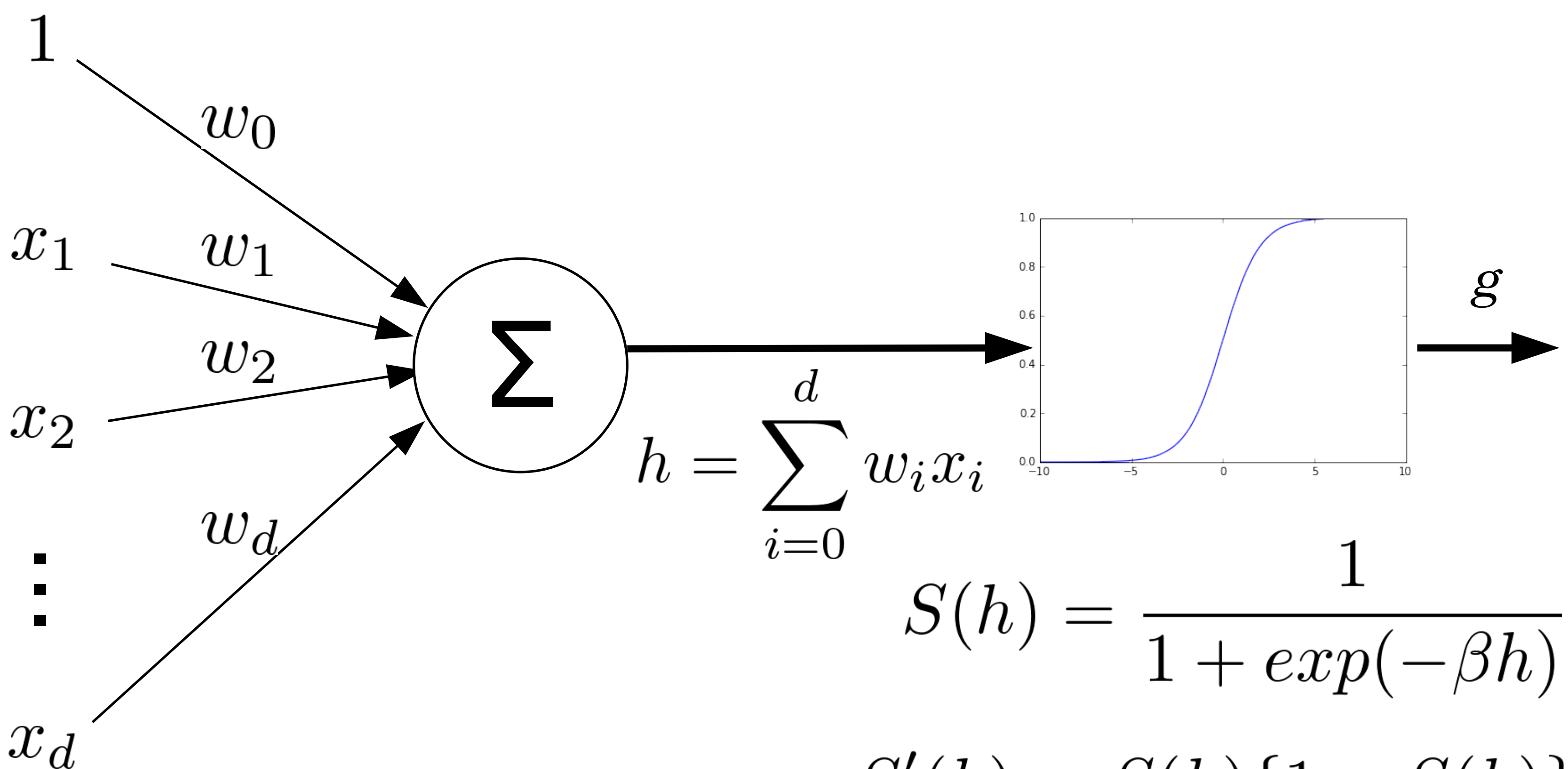
$$\frac{\partial J_p}{\partial g_{jp}} = g_{jp} - b_{jp}$$

- ユニット j が中間層の場合

$$\frac{\partial J_p}{\partial g_{jp}} = \sum_k \frac{\partial J_p}{\partial h_{kp}} \cdot \frac{\partial h_{kp}}{\partial g_{jp}} = \sum_k \varepsilon_{kp} w_{jk}$$

7.2 誤差逆伝播法による学習

- シグモイド関数の適用
  - 勾配計算の際に微分可能なものを用いる



$$h = \sum_{i=0}^{d} w_i x_i$$

$$S(h) = \frac{1}{1 + exp(-\beta h)}$$

$$S'(h) = S(h)\{1 - S(h)\}$$

7.2 誤差逆伝播法による学習

- 誤差の変化量

$$\varepsilon_{jp} = \begin{cases} (g_{jp} - b_{jp}) g_{jp} (1 - g_{jp}) & \text{出力層の場合} \\ (\sum_k \varepsilon_{kp} w_{jk}) g_{jp} (1 - g_{jp}) & \text{中間層の場合} \end{cases}$$

- 重みの修正式

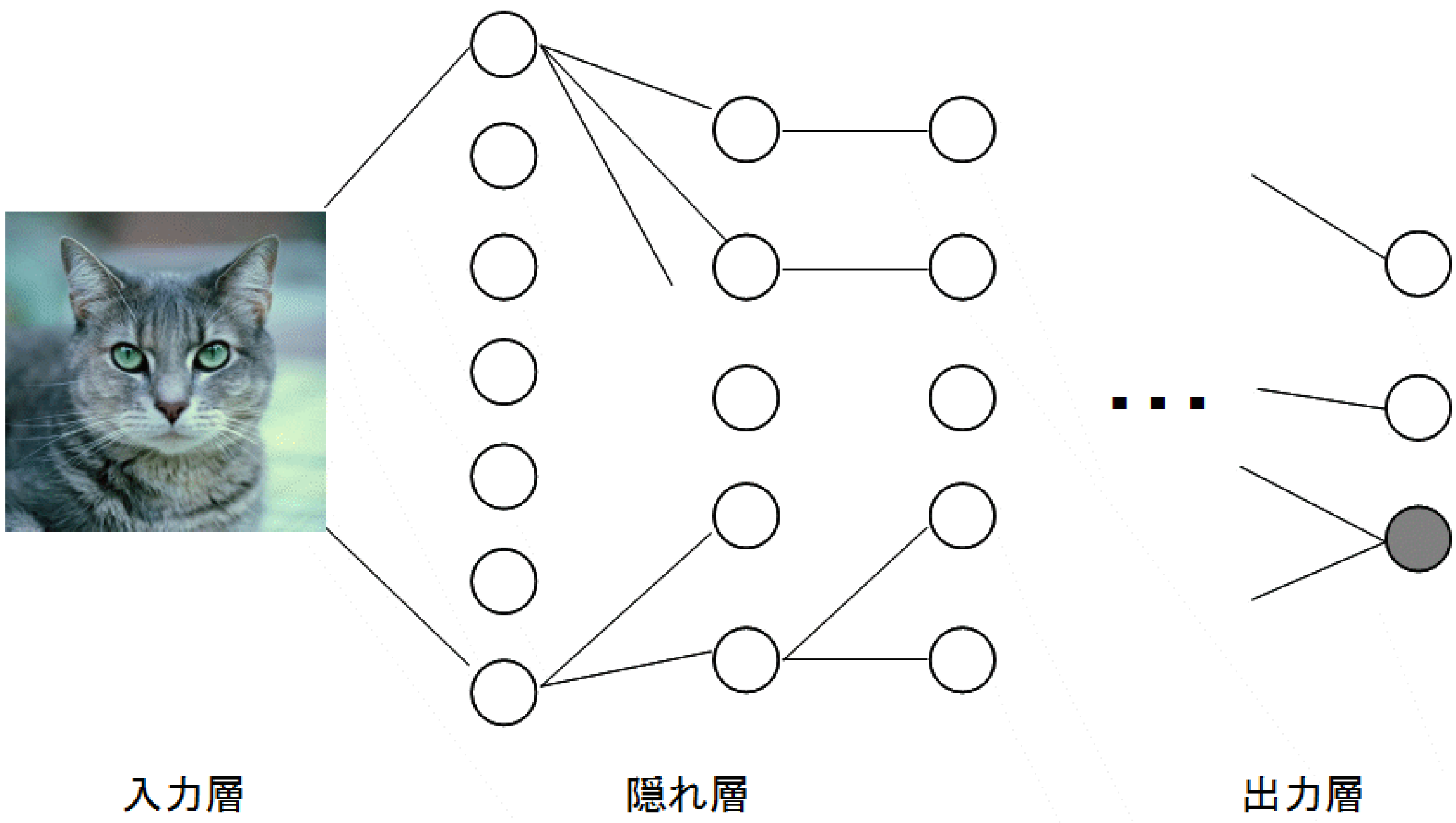
$$w'_{ij} = \begin{cases} w_{ij} - \rho (g_{jp} - b_{jp}) g_{jp} (1 - g_{jp}) g_{ip} & \text{出力層の場合} \\ w_{ij} - \rho (\sum_k \varepsilon_{kp} w_{jk}) g_{jp} (1 - g_{jp}) g_{ip} & \text{中間層の場合} \end{cases}$$

7.2 誤差逆伝播法による学習

- 過学習に気をつけよう
  - ニューラルネットワークは非線形識別面を学習することができるので、学習データの誤識別率を限りなく0に近づけることができる
  - そのような識別面は、未知データに対して誤識別率が高いことが多い
  - このように学習データに特化しすぎた識別面が学習される現象を過学習とよぶ

7.3 ディープニューラルネットワーク

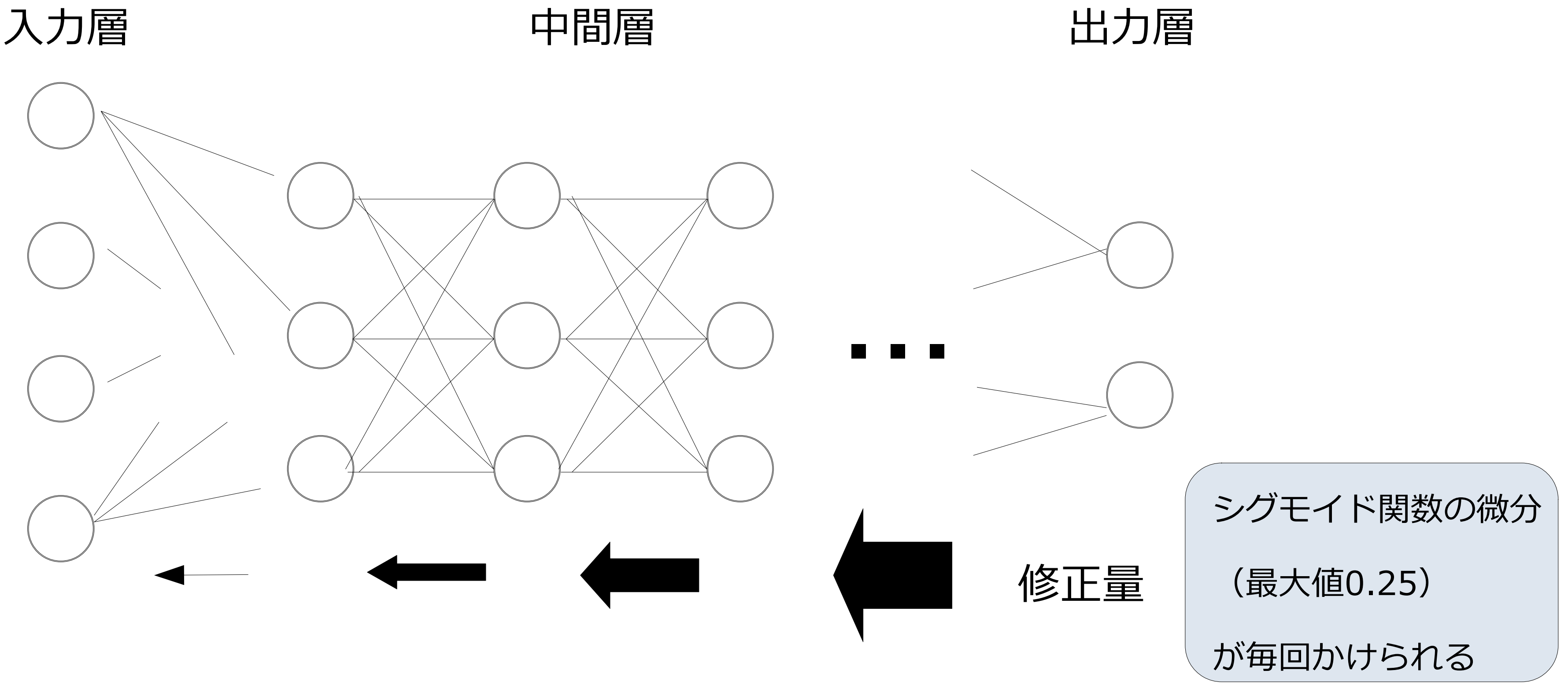
- 深層学習：多階層ニューラルネットによる学習
  - 多階層での学習を可能にする工夫
  - 問題に特化したネットワーク構造の導入



7.3.1 勾配消失問題とは

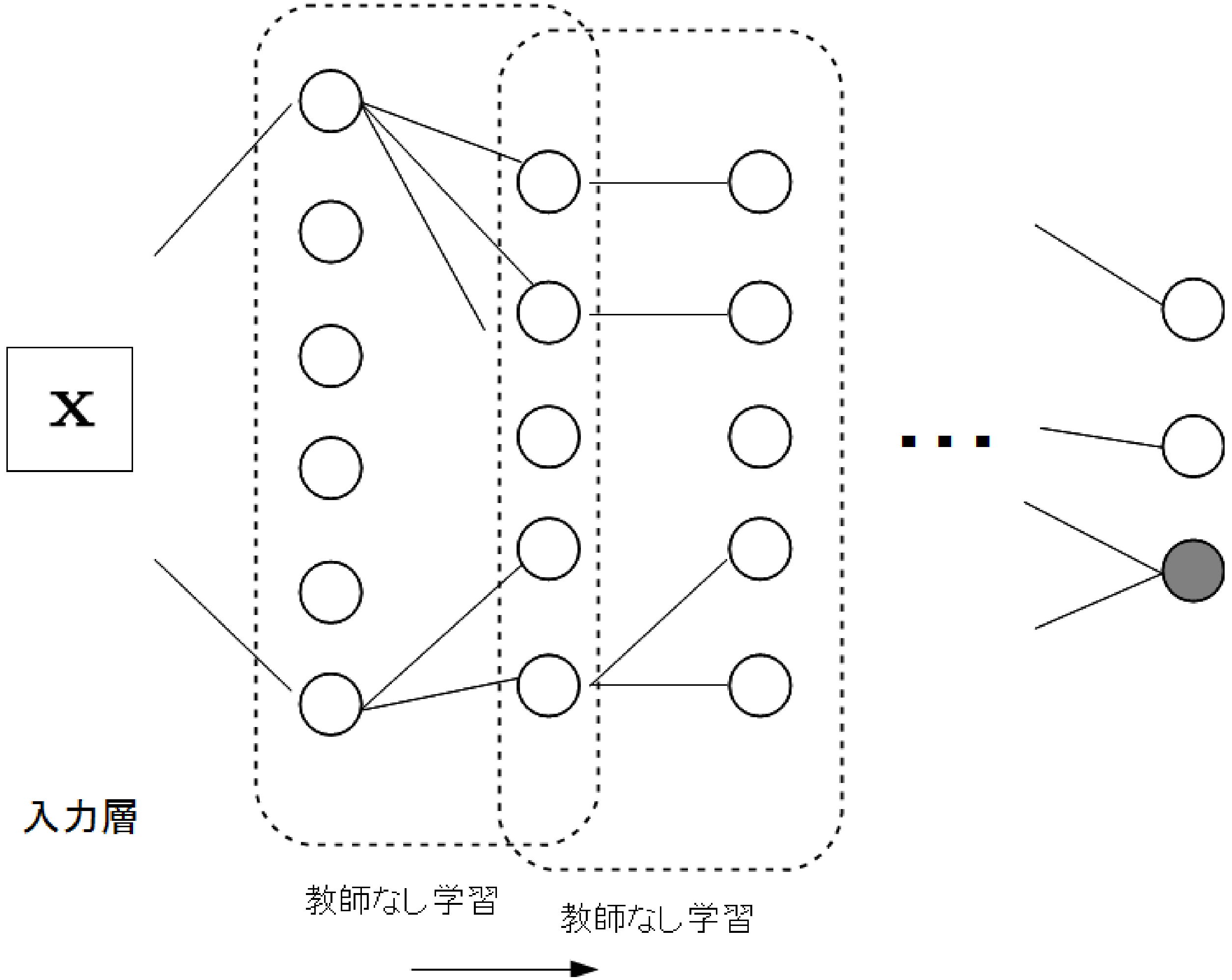
- 多階層における誤差逆伝播法の問題点
  - 入力層に近づくにつれて修正量が消失する

順方向：非線形　　　逆方向：線形



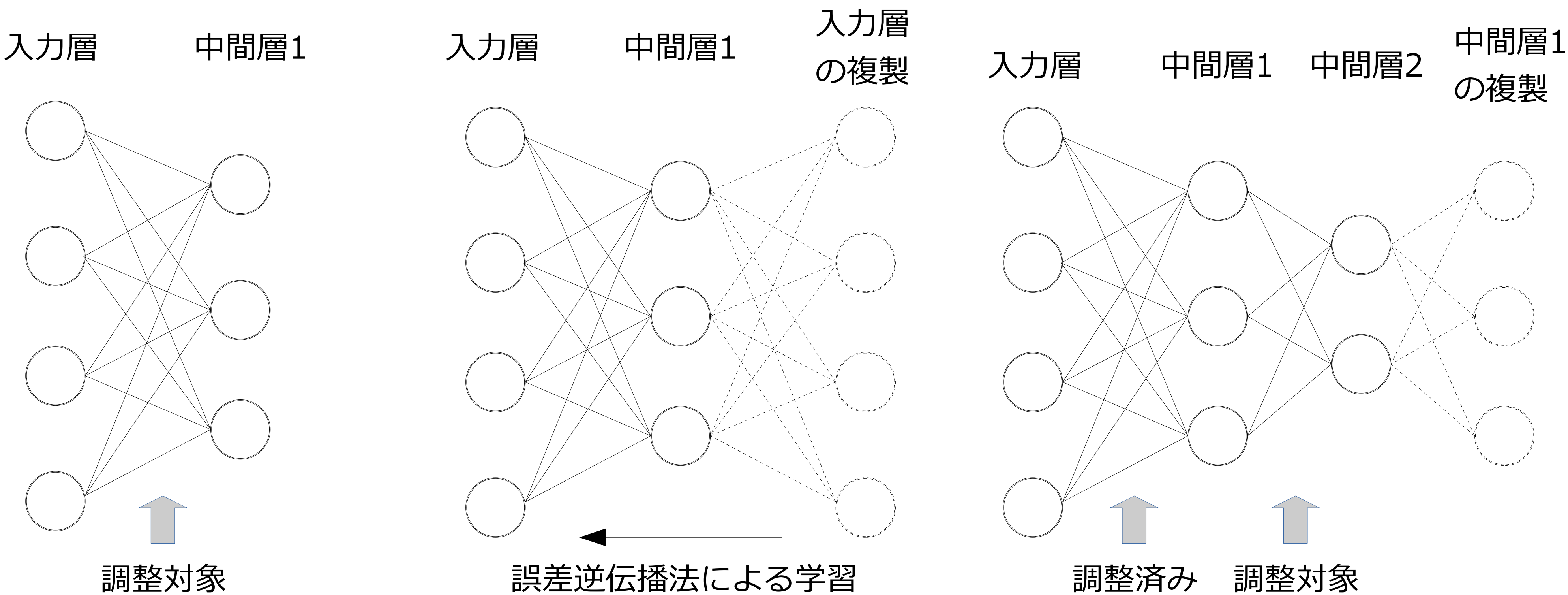
7.3.2 多階層学習における工夫

- 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

- 事前学習法のアイディア
  - 自己写像学習による重みの初期値設定

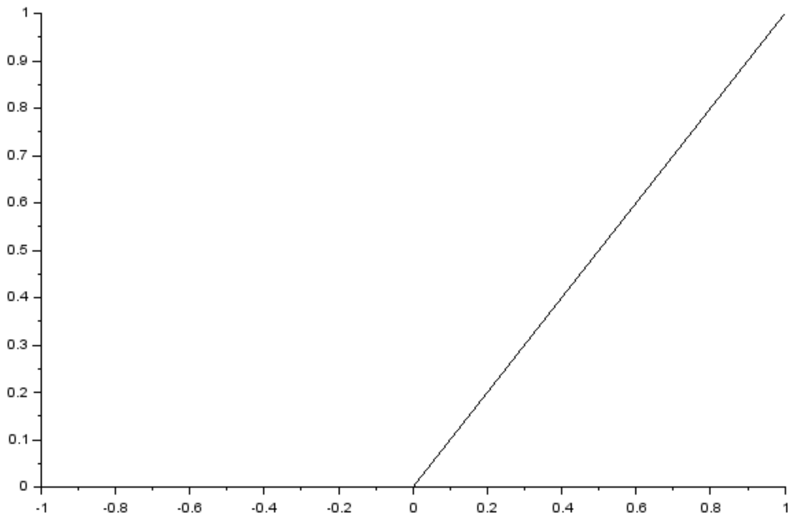


(a) 事前調整対象の重み　(b) オートエンコーダによる復元学習　(c) 1階層上の事前調整

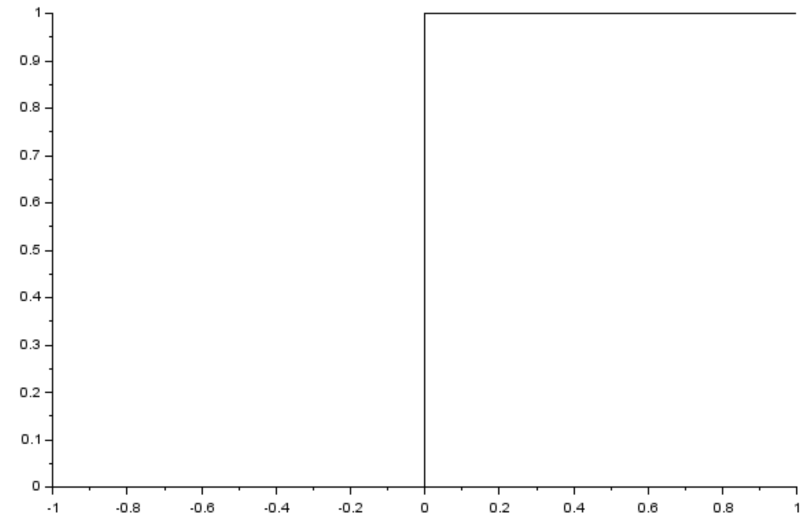
7.3.2 多階層学習における工夫

- 活性化関数をrectified linear関数(ReLU)に変更

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



(a) rectified linear 関数

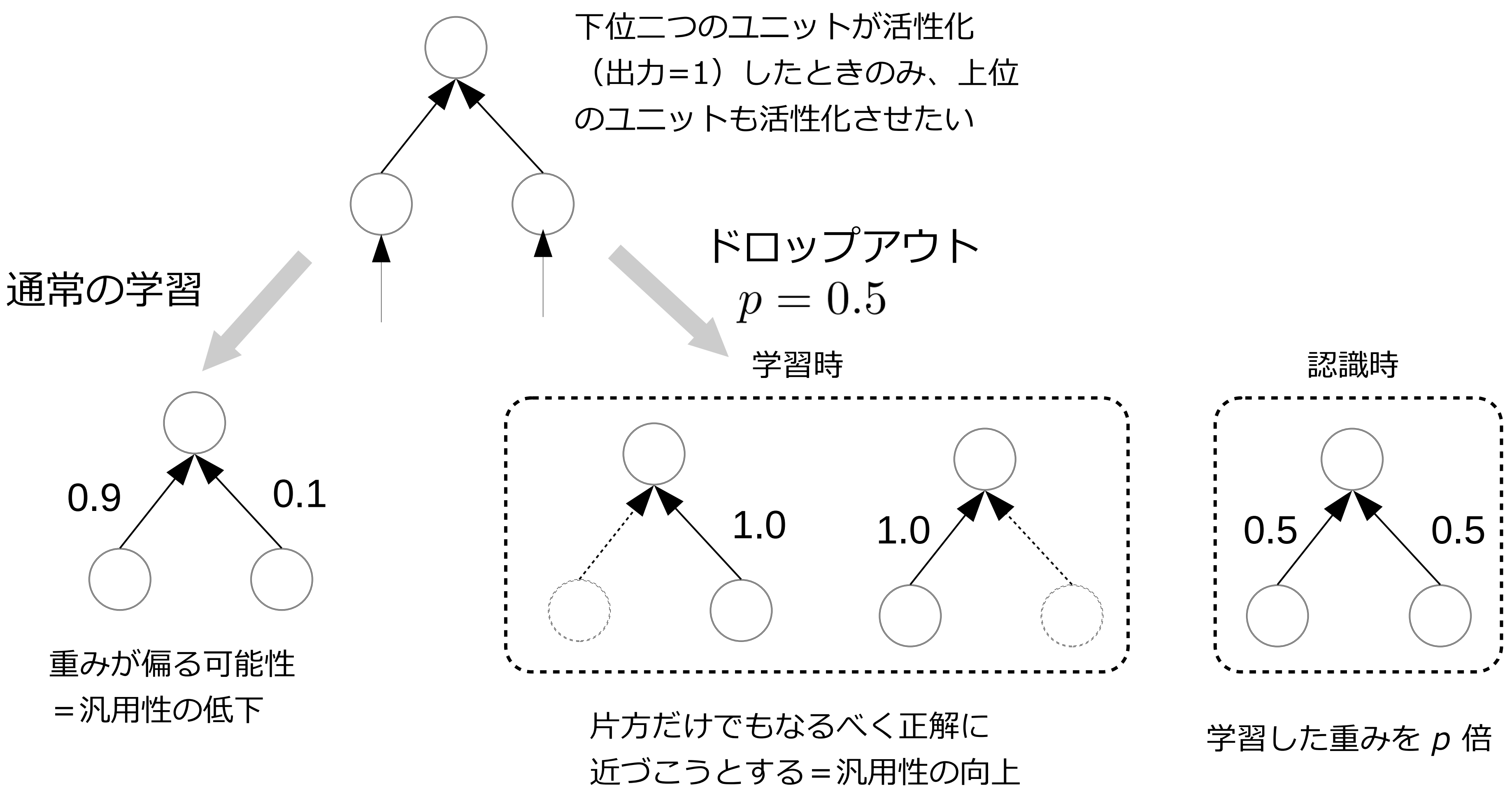


(b) (a)の導関数

- 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

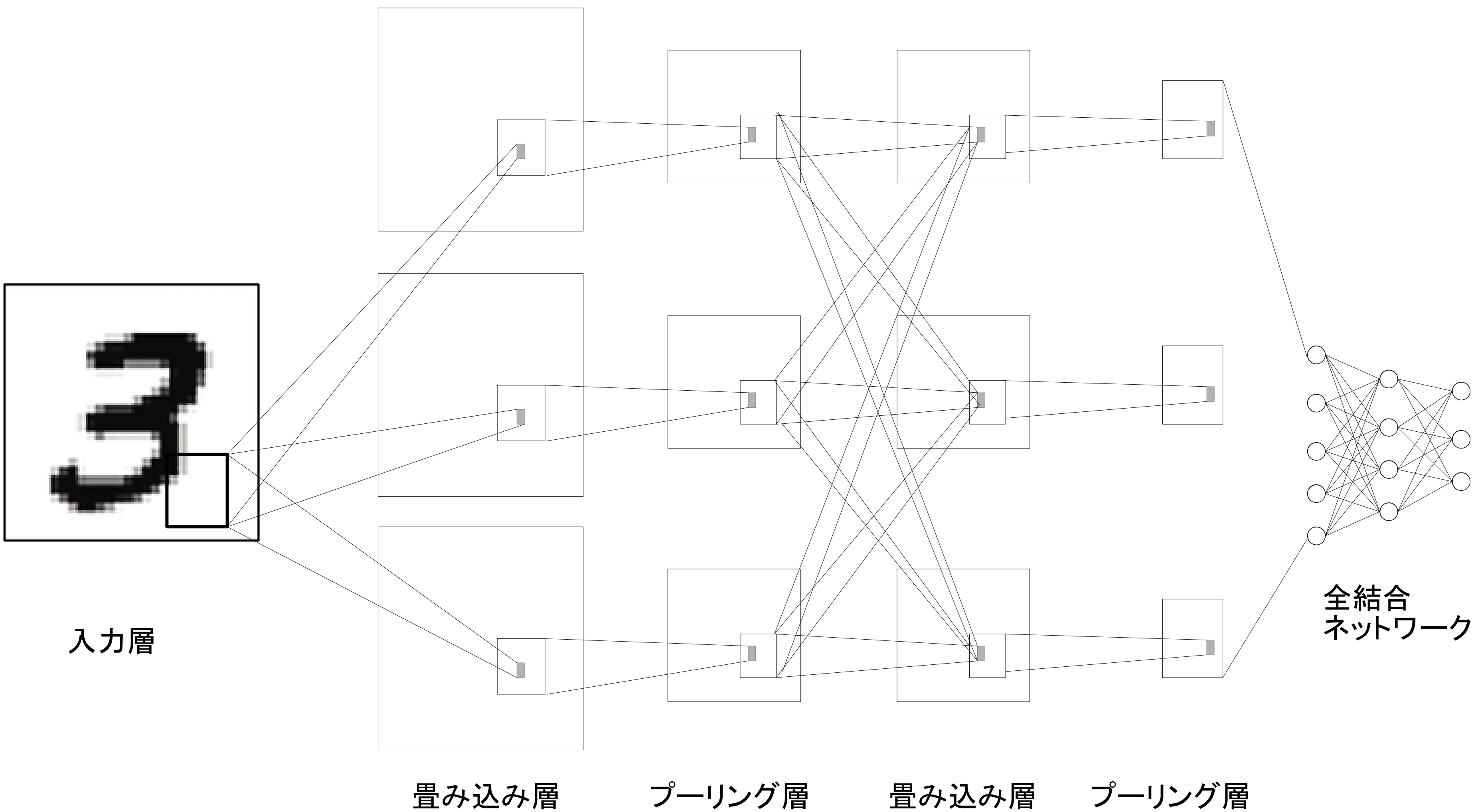
7.3.2 多階層学習における工夫

- 過学習の回避
  - ドロップアウト：ランダムに一定割合のユニットを消して学習を行う



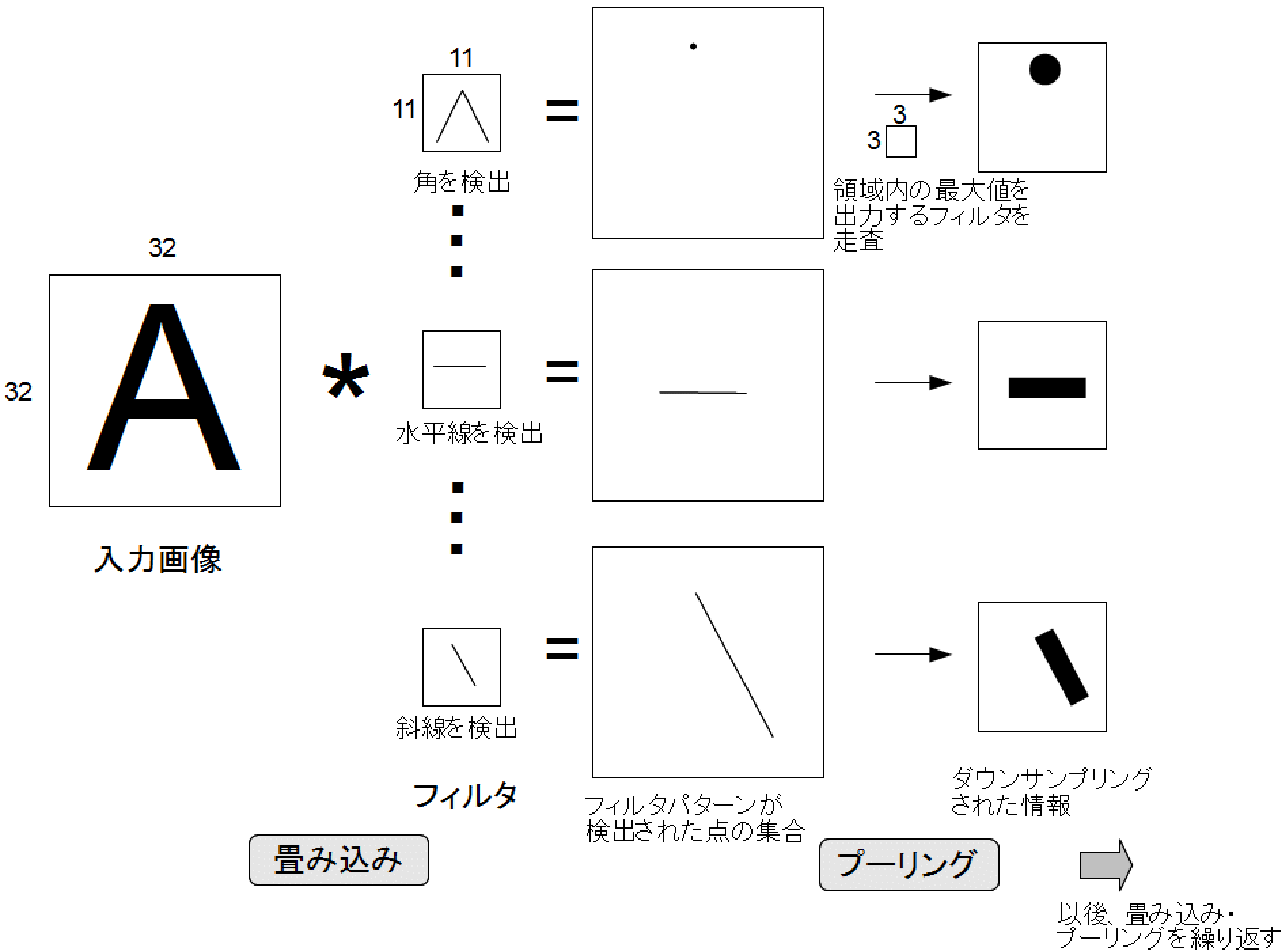
7.3.3 特化した構造をもつニューラルネットワーク

- 畳み込みニューラルネットワーク
  - 画像認識に適するネットワーク構造



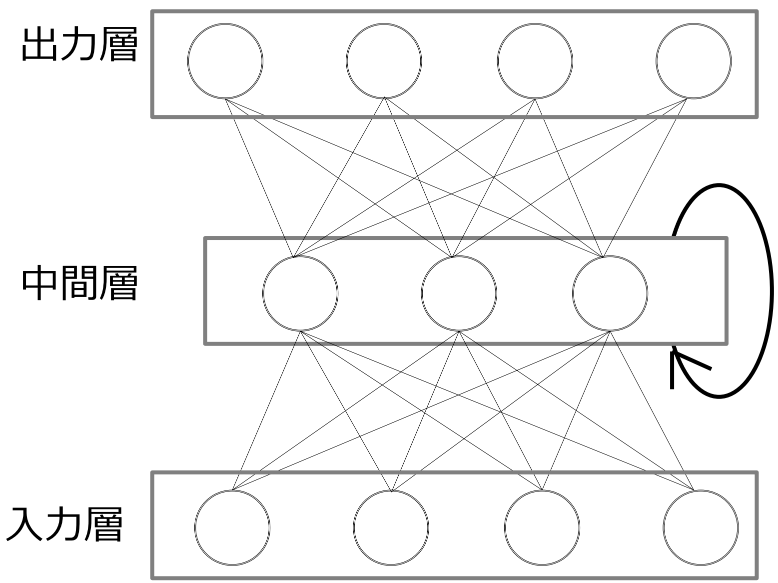
7.3.3 特化した構造をもつニューラルネットワーク

- 畳み込みニューラルネットワークの演算

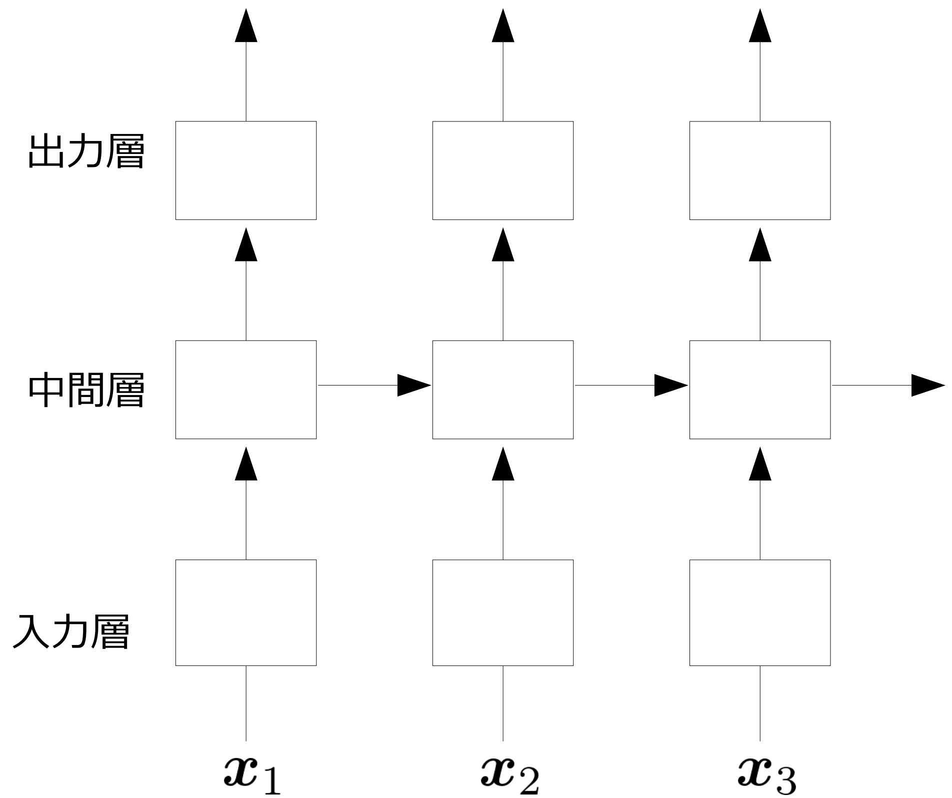


7.3.3 特化した構造をもつニューラルネットワーク

- 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