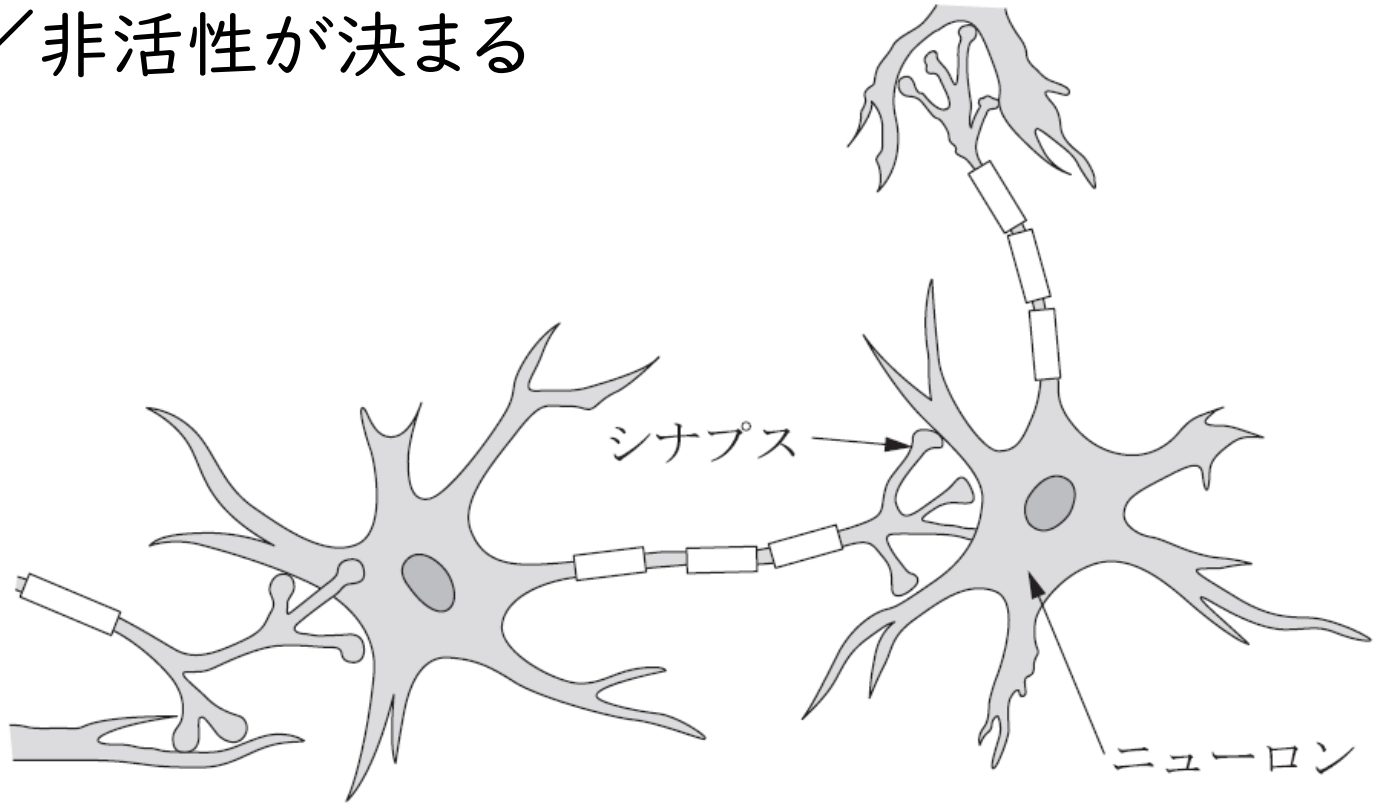
7. 限界は破れるか（2）－ ニューラルネットワーク －

- 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、どのような非線形関数にするかを事前に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

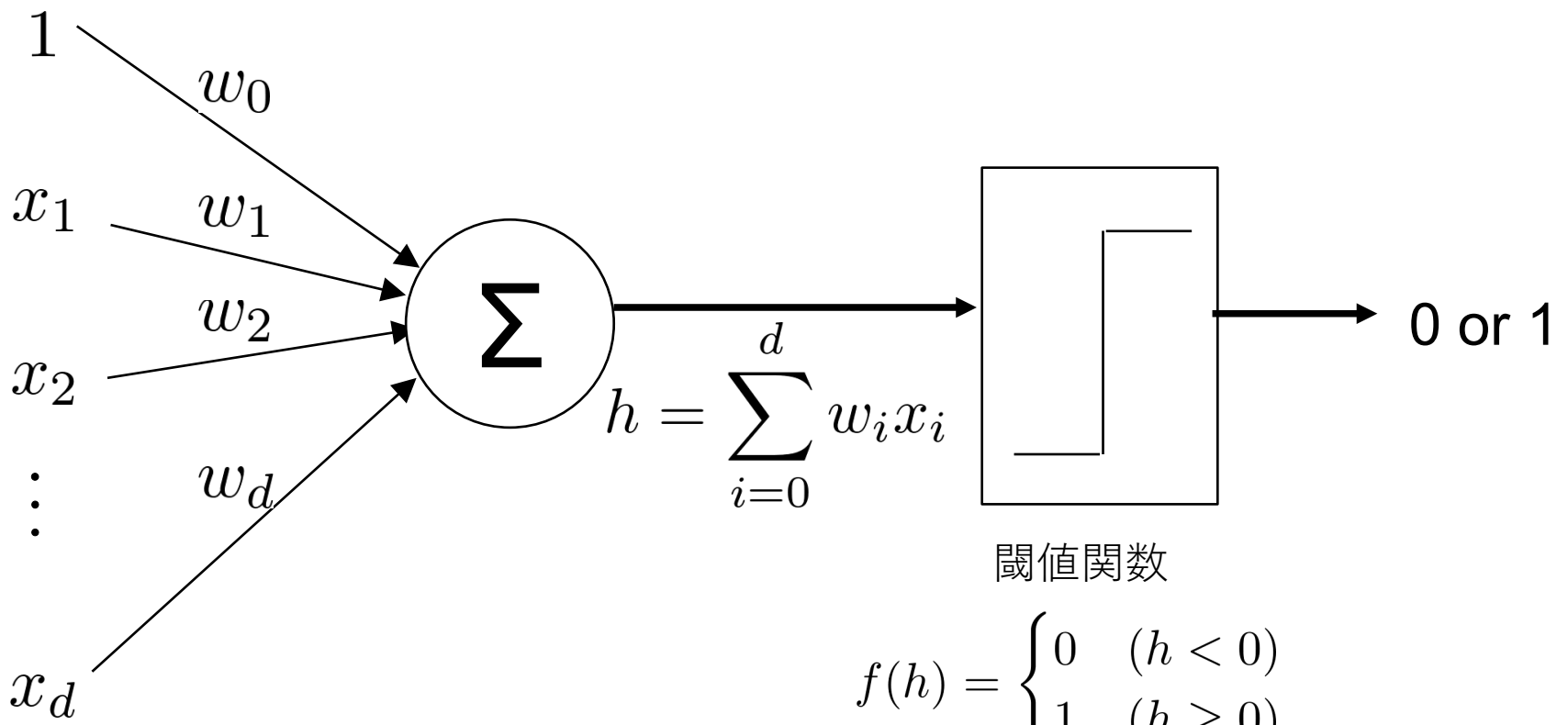
- 神経細胞の計算メカニズム
  - ニューロンがシナプス結合によって複雑に結合
  - 入力された電気信号の重み付き和の値によって、各ニューロンの活性／非活性が決まる



7.1 ニューラルネットワークの構成

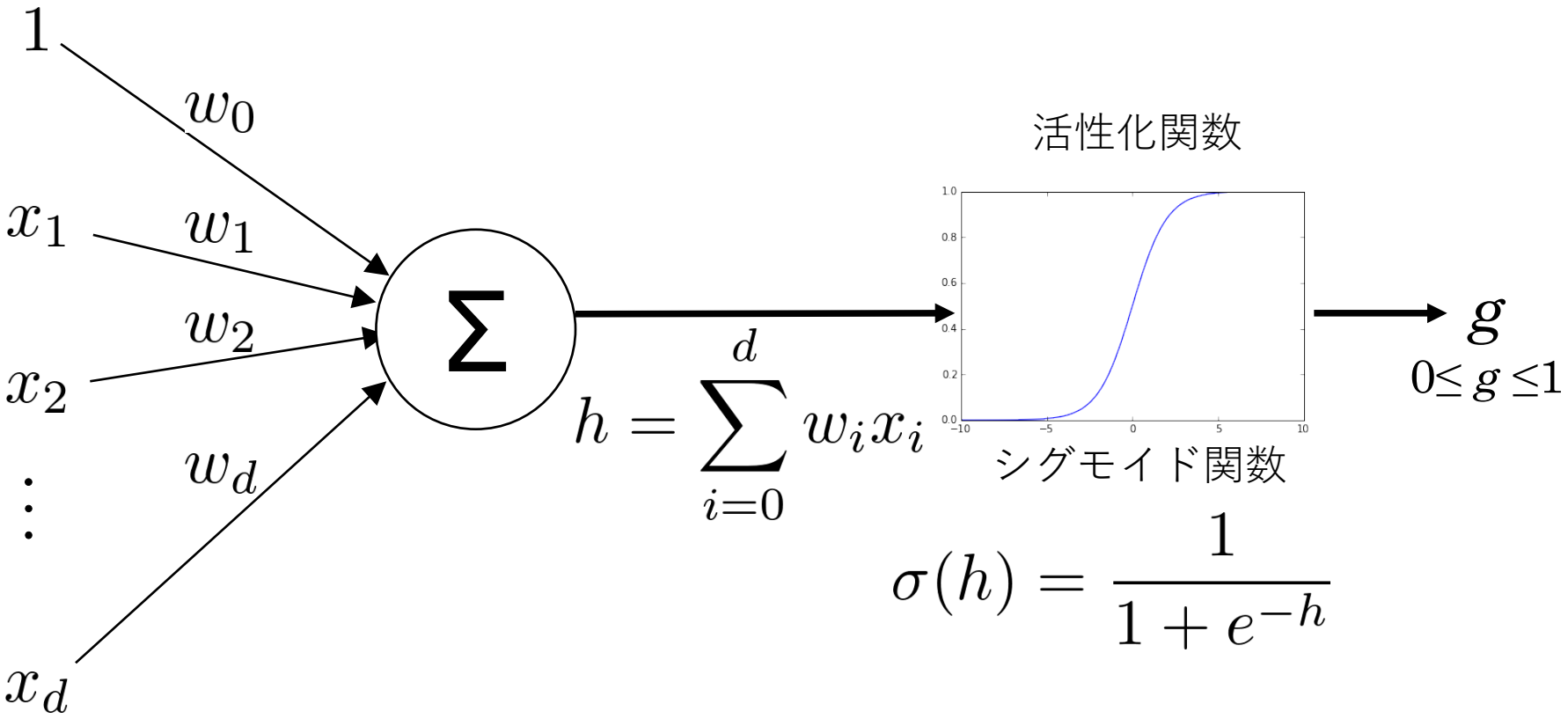
- 閾値論理ユニットによるニューロンのモデル化
  - ◆ $\boldsymbol{w}^T\boldsymbol{x} = 0$ という特徴空間上の識別面を表現（ $\boldsymbol{w}$, $\boldsymbol{x}$はd+1次元）

    → パーセプトロン（データが線形分離可能なときのみ学習可能）



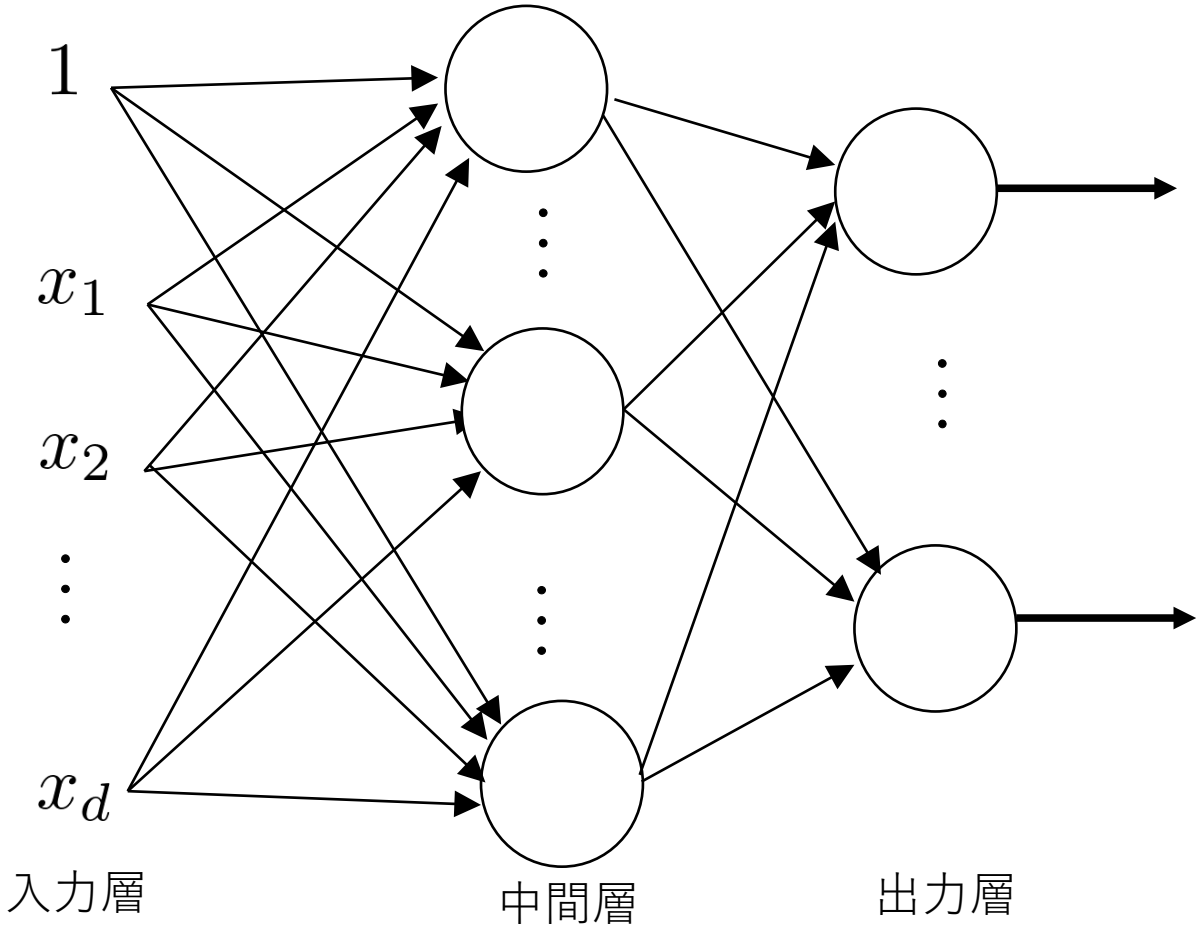
7.1 ニューラルネットワークの構成

- 活性化関数をシグモイド関数に差し替え
  - 入力の重み付き和を大小関係は変えずに0～1の値に変換
  - データが線形分離不可能でも誤差最小の線形識別面が学習可能



7.1 ニューラルネットワークの構成

- フィードフォワード型ニューラルネットワーク
  - ユニットを階層状に結合することで非線形識別面を表現



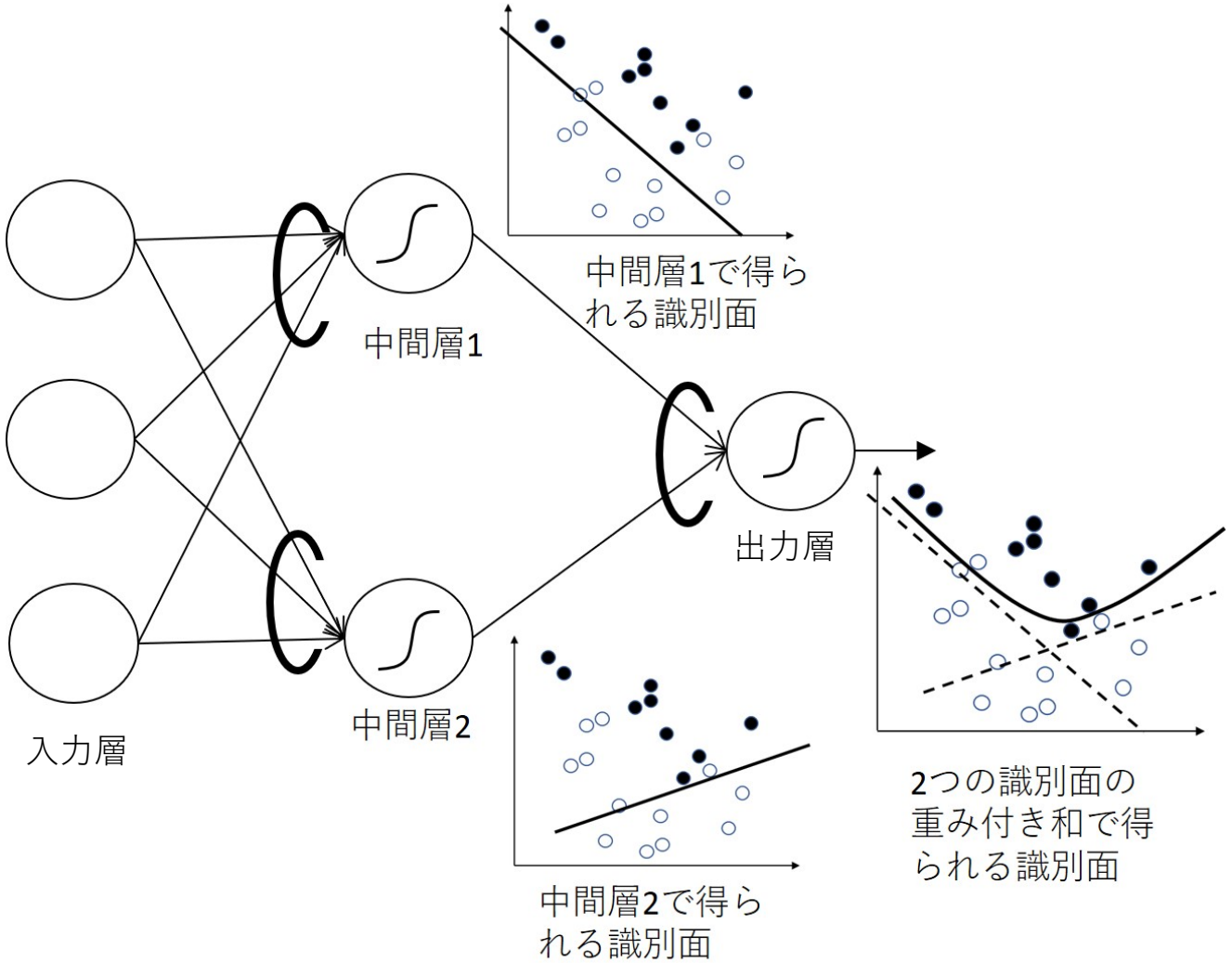
多クラス識別の出力層には
活性化関数として以下の
softmax関数を用いる

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

2クラスの場合はシグモイド関数と同じ

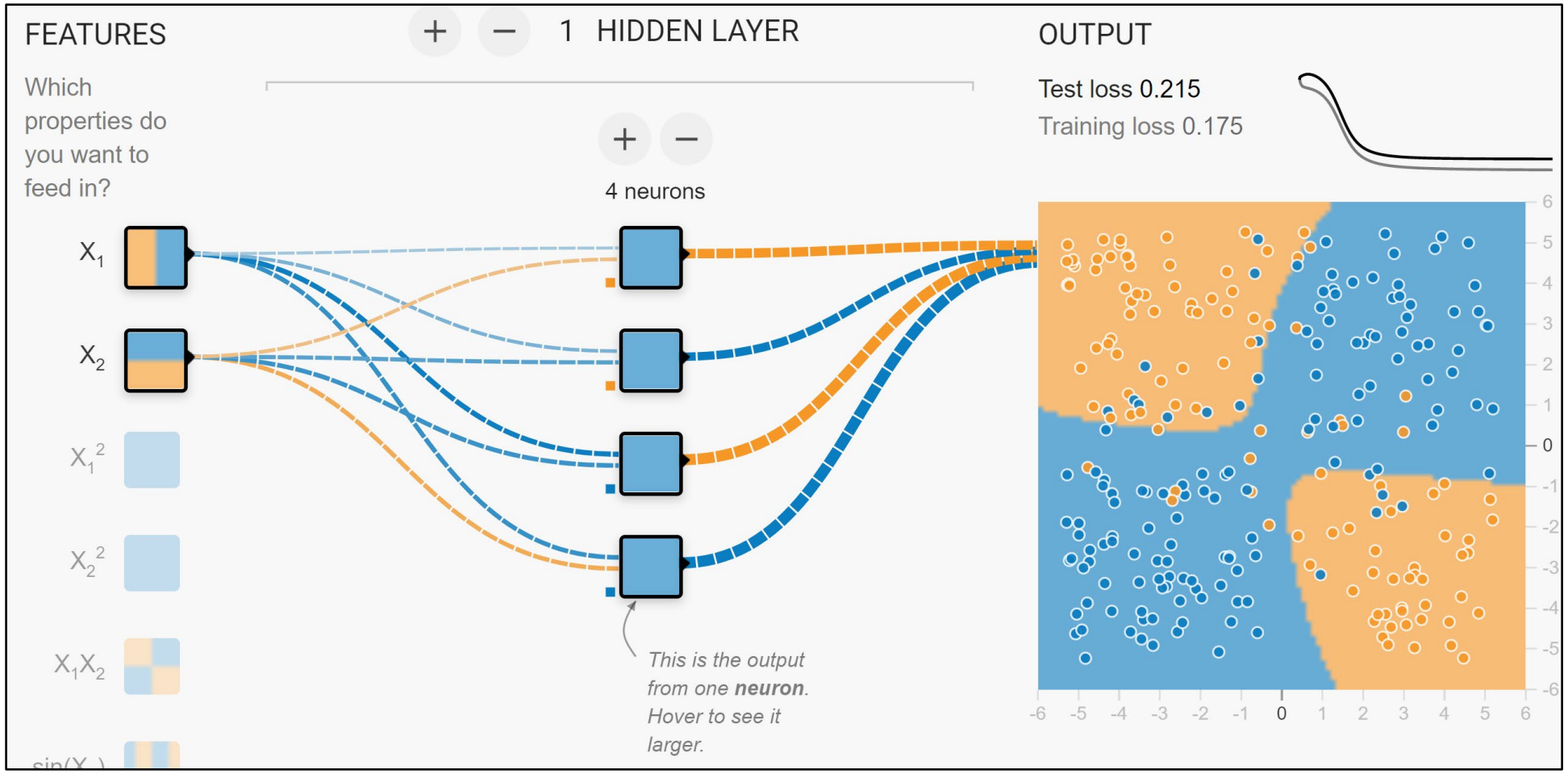
7.1　ニューラルネットワークの構成

- ニューラルネットワークによる非線形識別面の実現



7.1 ニューラルネットワークの構成

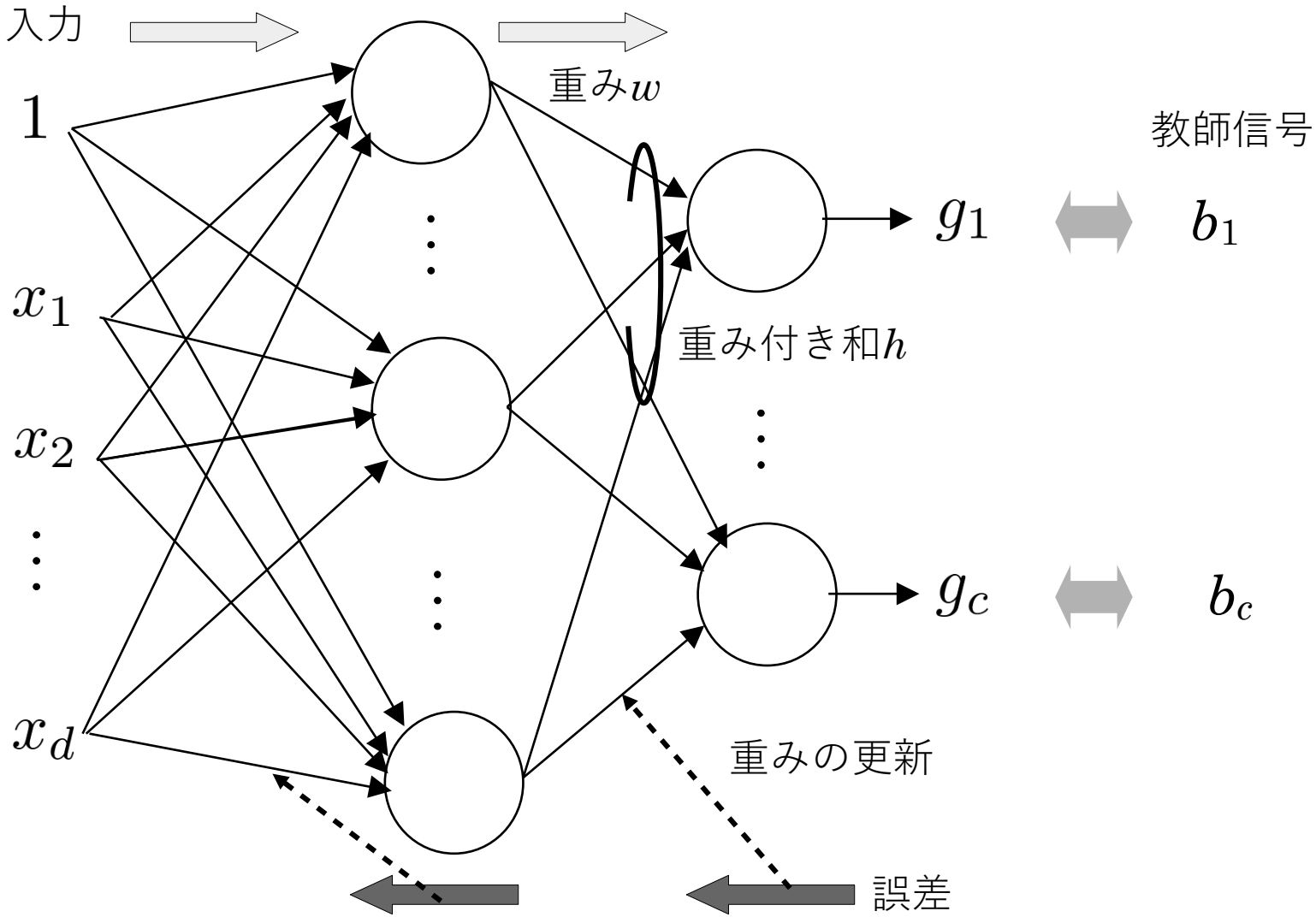
- ニューラルネットワークによる非線形識別面の実現



https://playground.tensorflow.org/

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

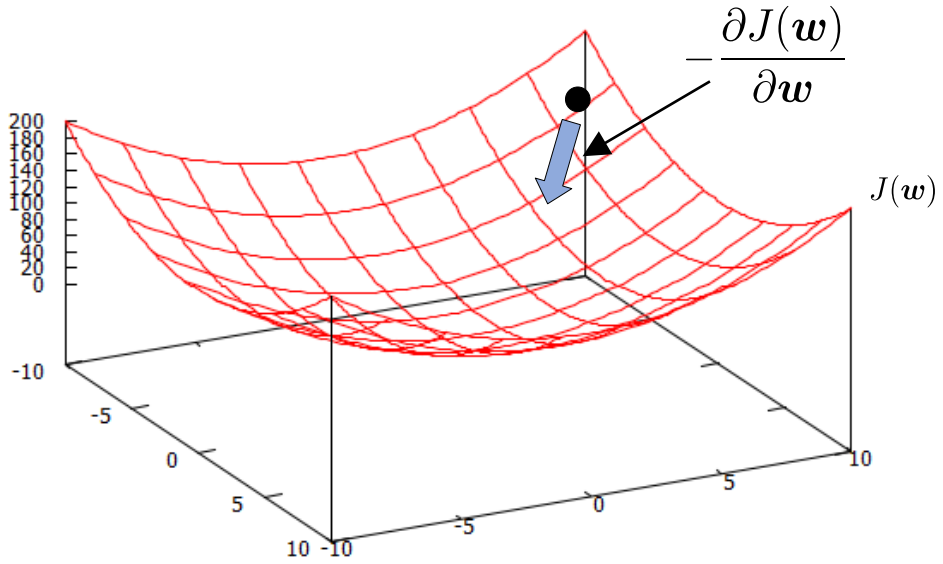


7.2 誤差逆伝播法による学習

- 結合重みの調整アルゴリズム
  - 特定のデータ $\boldsymbol{x}_p$ に対する二乗誤差
  - J は $\boldsymbol{w}$ の関数

$$J(\boldsymbol{w}) \equiv \frac{1}{2}\sum_{i=1}^{c}(g_i(\boldsymbol{x}_p) - b_i)^2$$

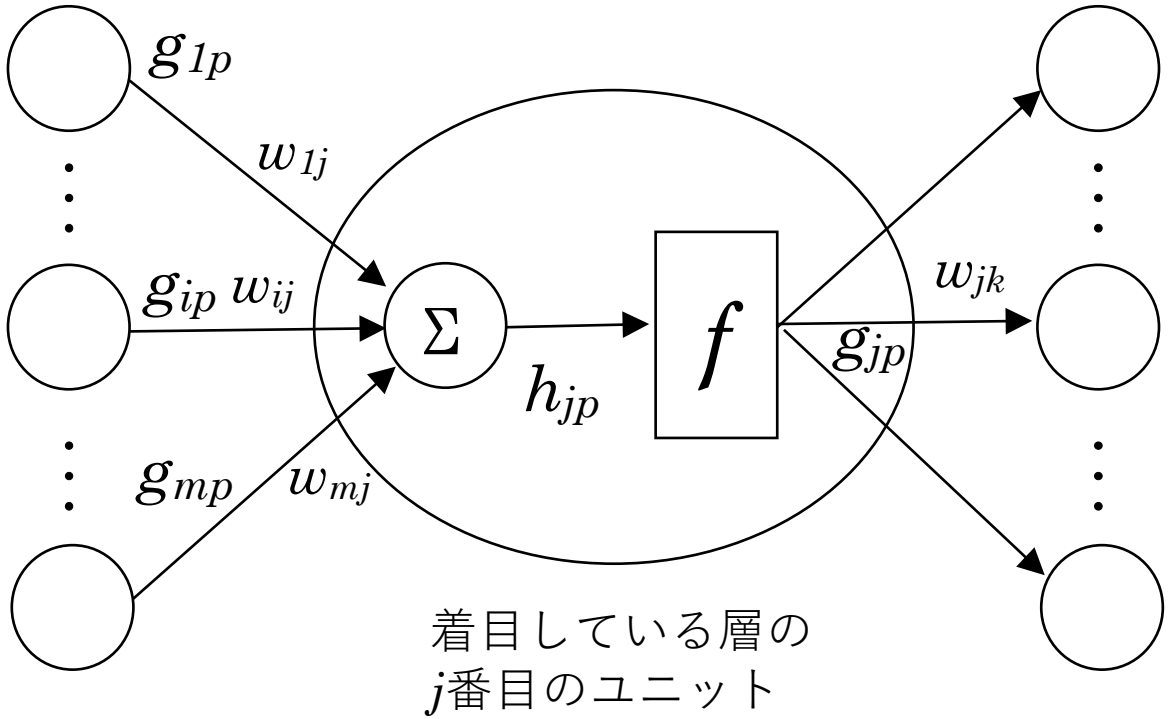
    - $\boldsymbol{w}$ を J の勾配方向へ一定量だけ動かすことを繰り返して、最適解へ収束させる（→勾配降下法）



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

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

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

7.2 誤差逆伝播法による学習

- 調整量の計算
  - 合成関数の微分の公式を使う

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

- 右辺第2項は g_{ip}
- 右辺第1項を ε_{jp} とおく

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

7.2 誤差逆伝播法による学習

- 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

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

- 活性化関数の微分
  - シグモイド関数の場合：$g_{jp}\,(1\text{-}g_{jp})$

7.2 誤差逆伝播法による学習

- 誤差の変化量

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

- 重みの修正式

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

7.2 誤差逆伝播法による学習

- 誤差逆伝播法

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 $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ に対してエポック数だけ以下繰り返し
   - 入力 $\boldsymbol{x}_i$ に対するネットワークの出力 $\boldsymbol{g}_i$ を計算

   a. 出力層のユニットに対してエラー量計算

   b. 中間層のユニットに対してエラー量計算

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

   c. 重みの更新

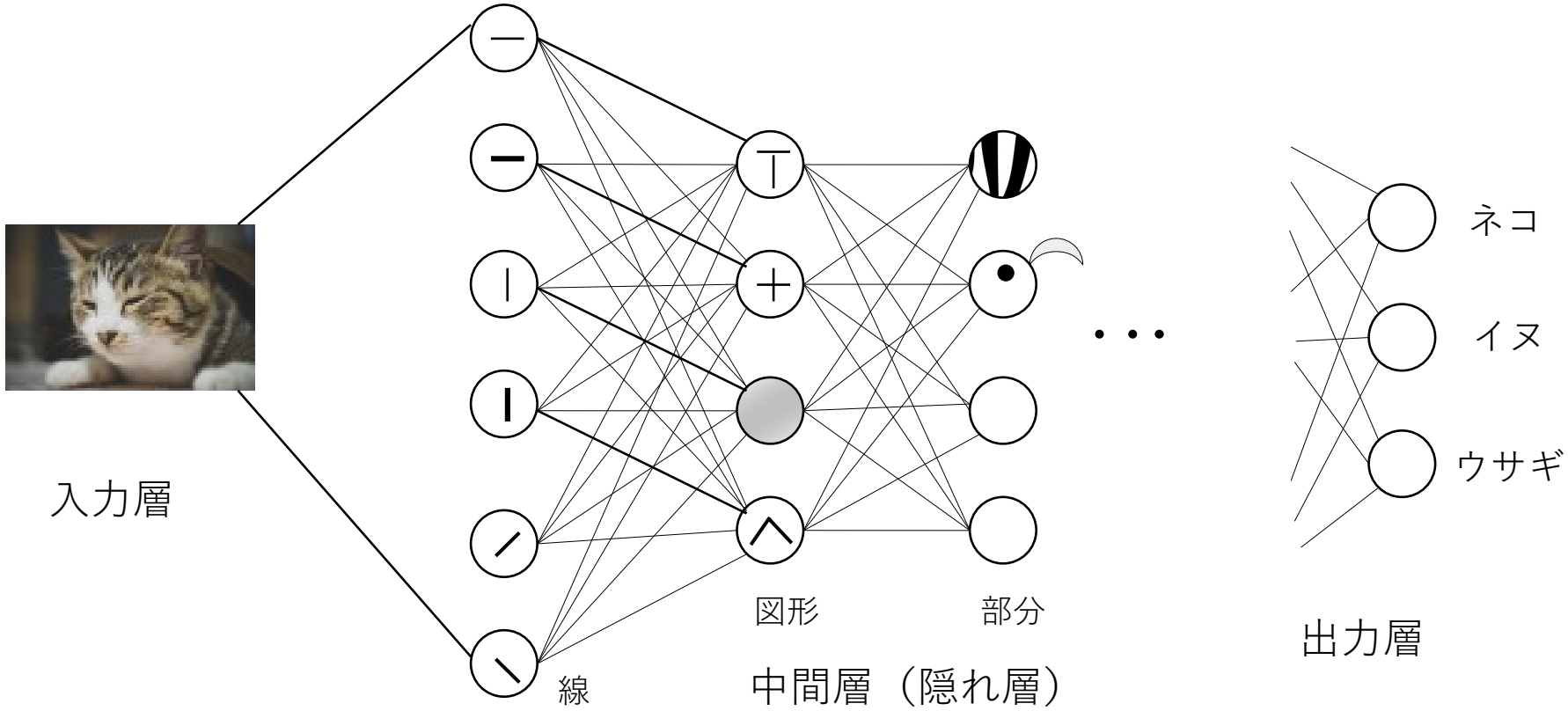
$$w'_{ij} = \begin{cases} w_{ij} - \rho(g_{jp} - b_{jp}) g_{jp}(1 - g_{jp}) g_{ip} & \text{出力層} \\ w_{ij} - \rho(\sum_k \varepsilon_{kp} w_{jk}) g_{jp}(1 - g_{jp}) g_{ip} & \text{中間層} \end{cases}$$

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
- 例題7.1

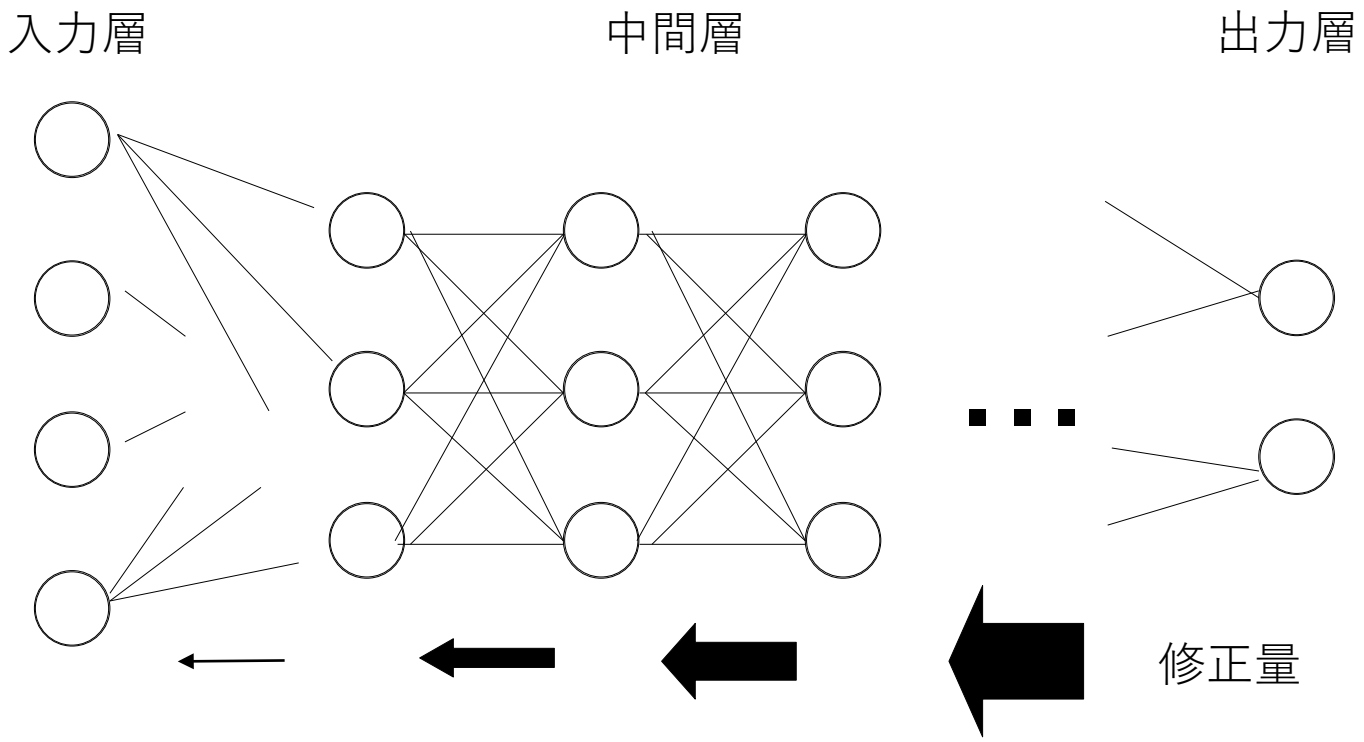
7.3 ディープニューラルネットワーク

- 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

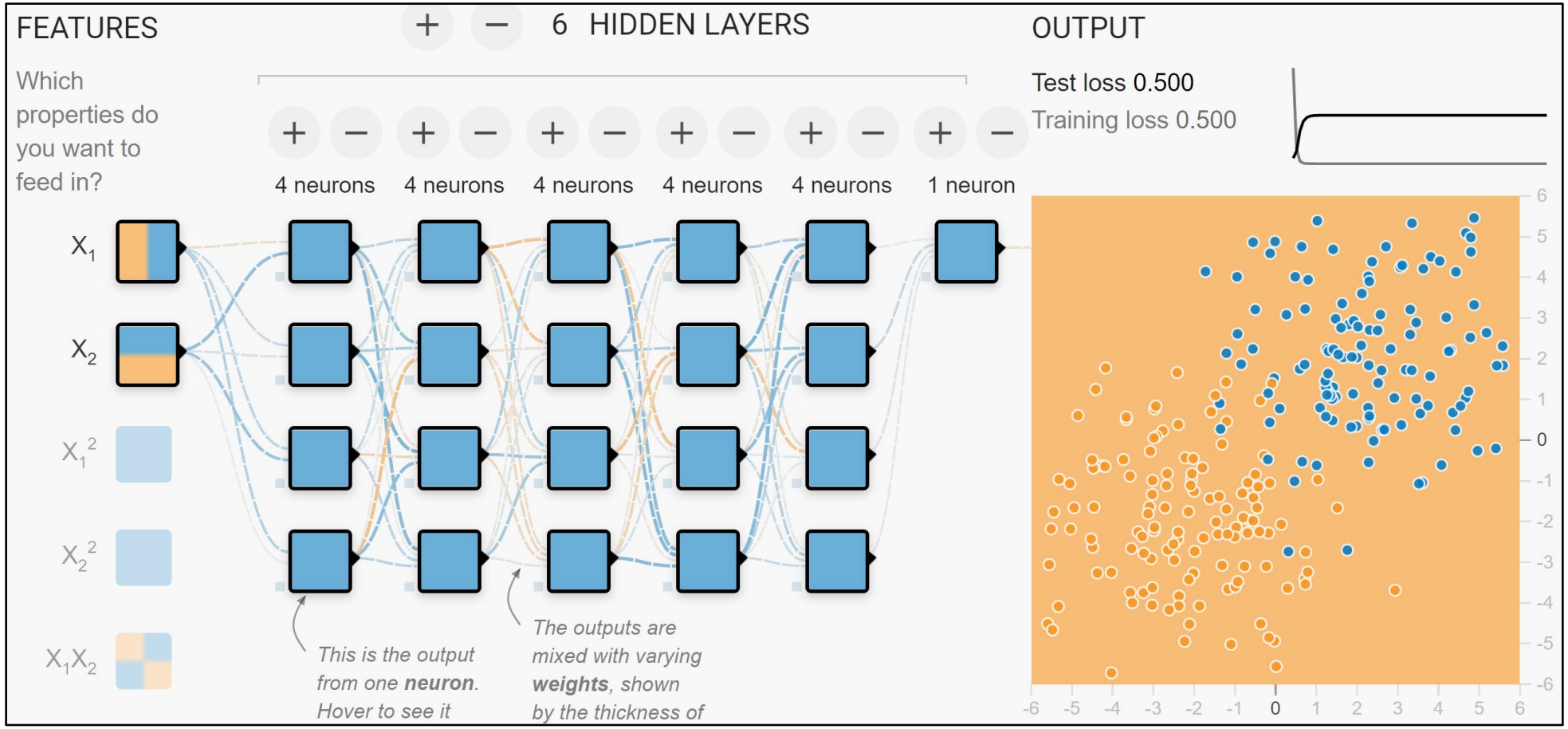


7.3.1 勾配消失問題とは

- 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



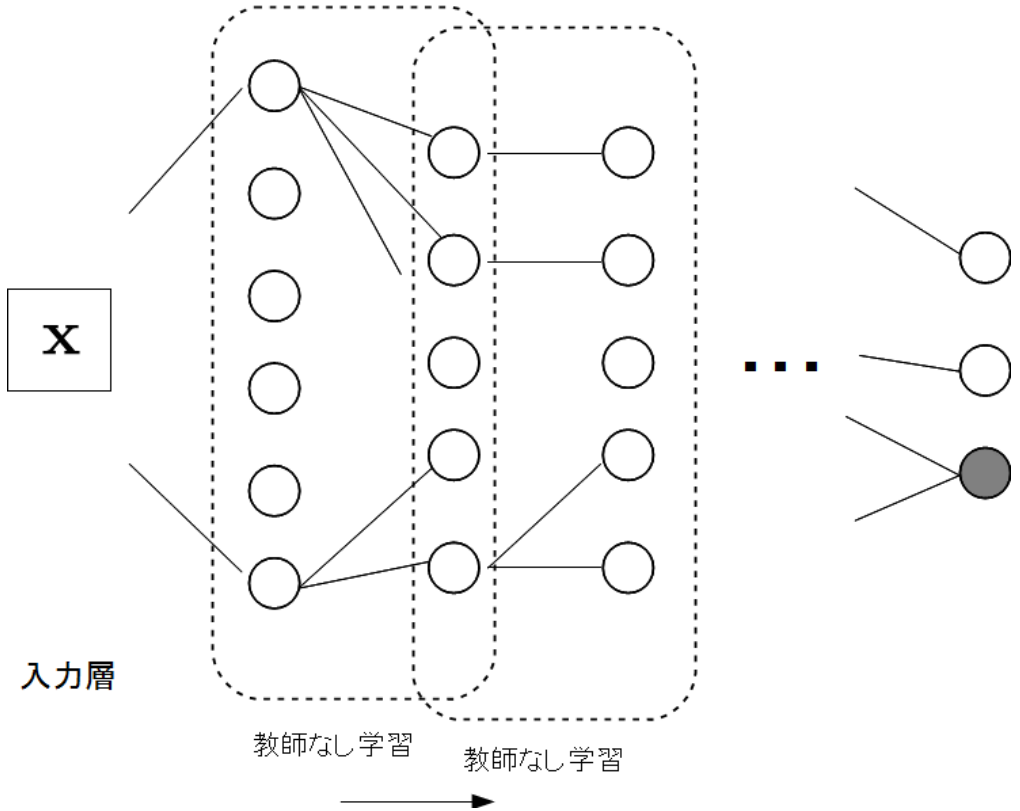
7.3.1 勾配消失問題とは



https://playground.tensorflow.org/

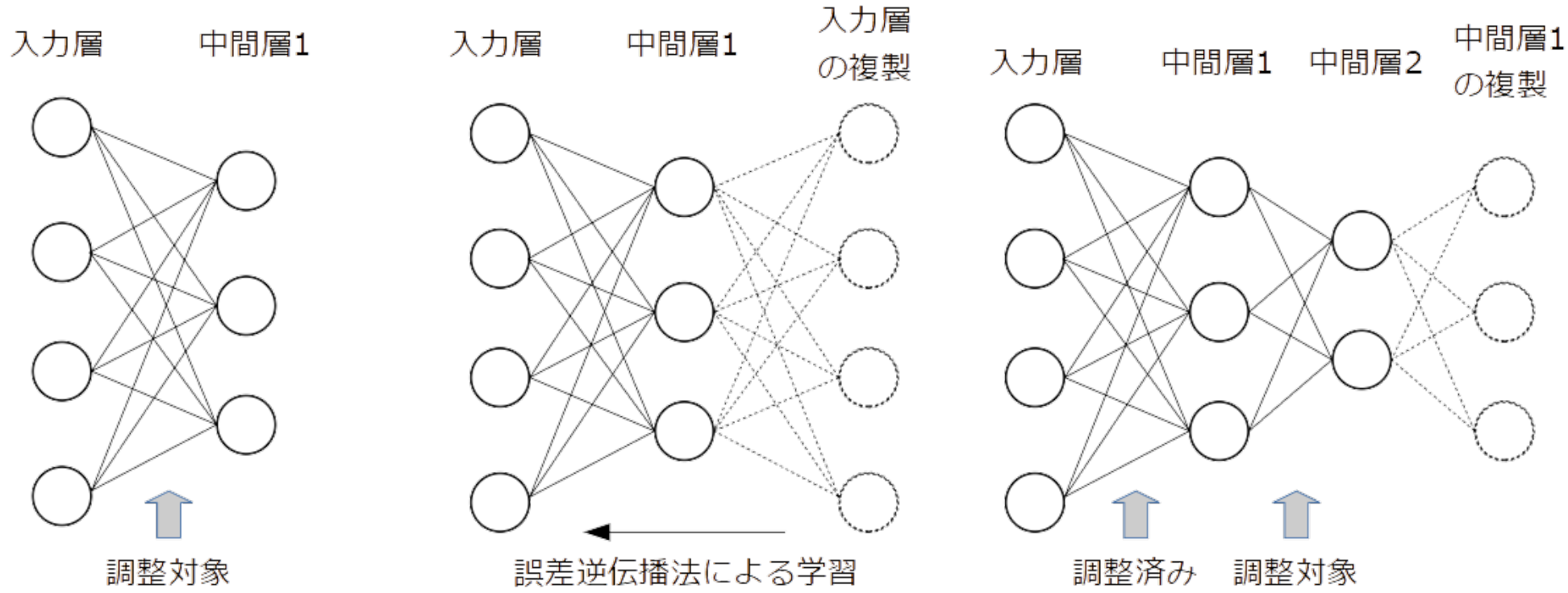
7.3.2 多階層学習における工夫

- 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

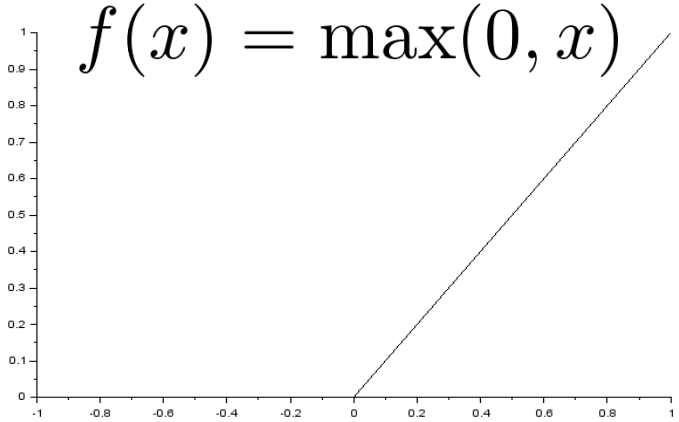
- 事前学習法のアイディア
  - 自己写像学習による重みの初期値設定



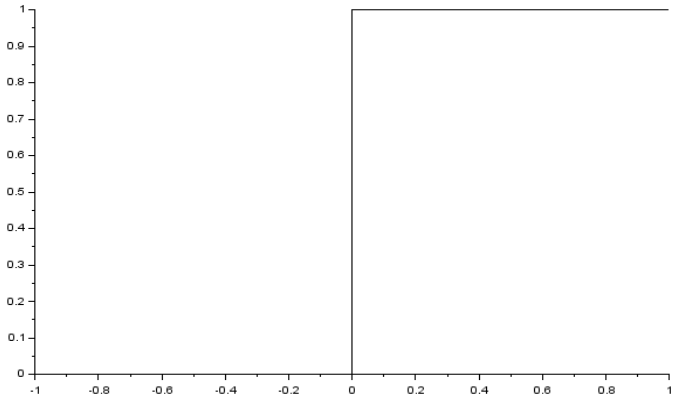
(a) 事前調整対象の重み　(b) オートエンコーダによる復元学習　(c) 1階層上の事前調整

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
- 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



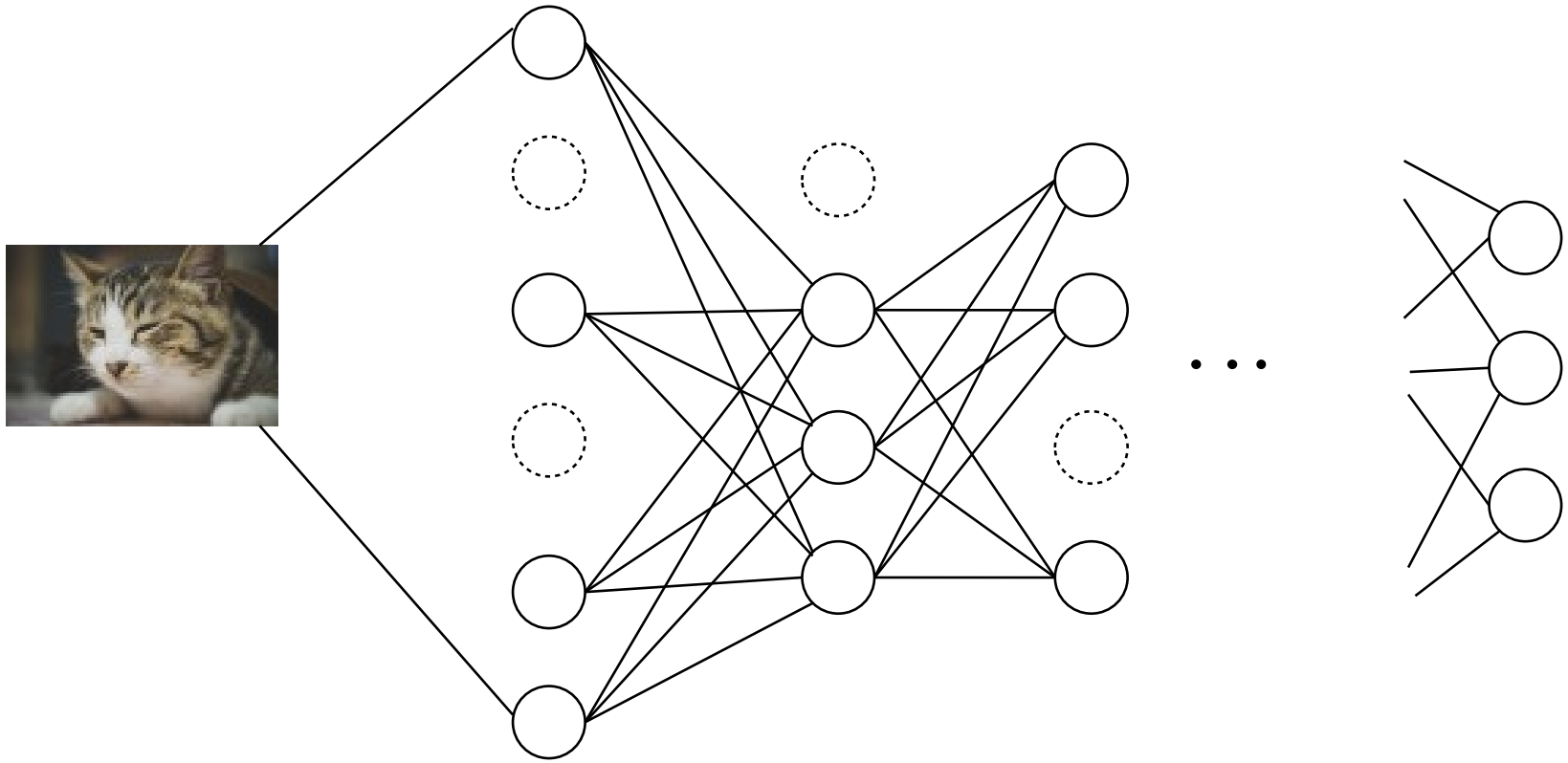
(a) rectified linear 関数



(b) (a)の導関数

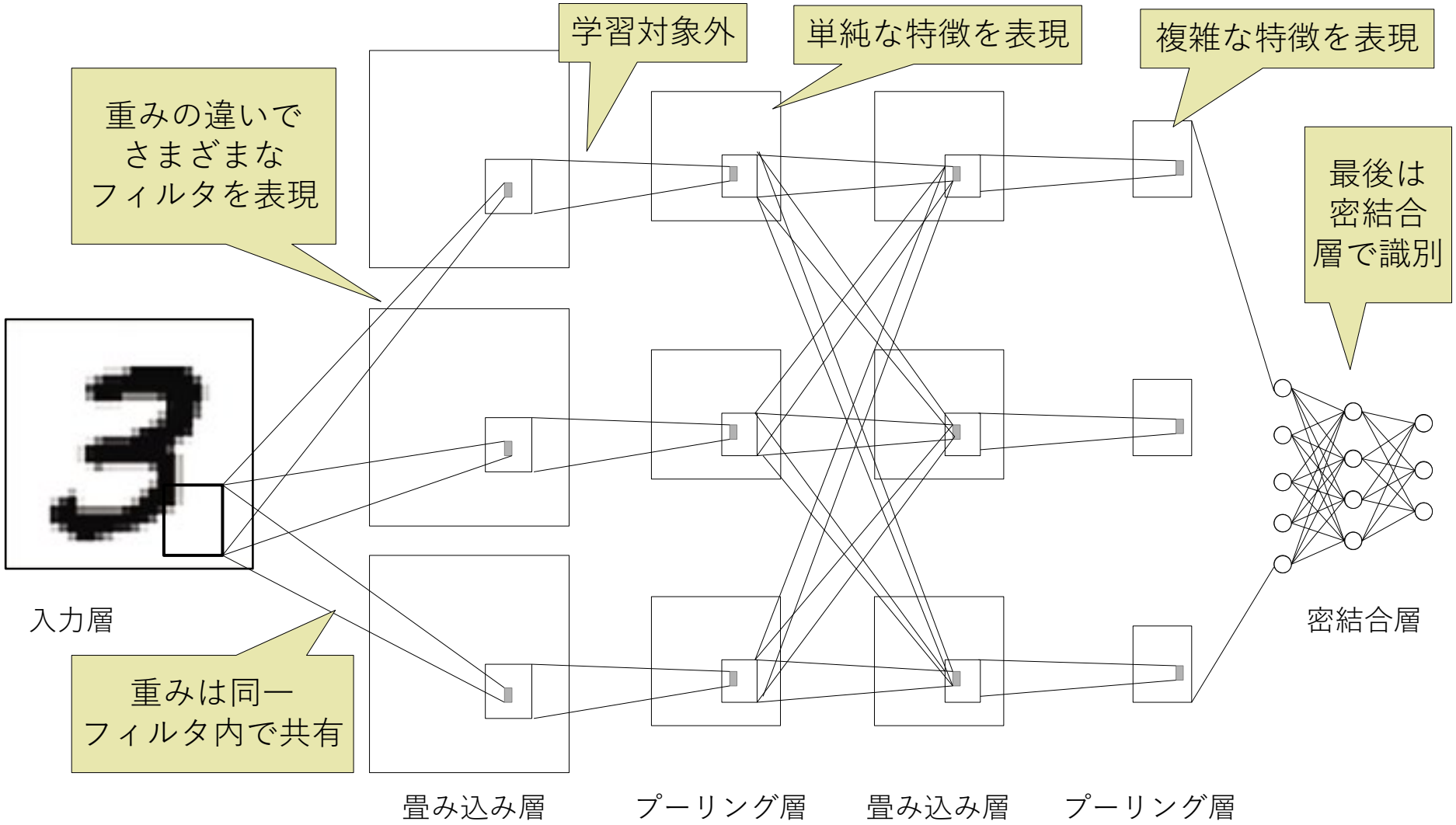
7.3.2 多階層学習における工夫

- 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



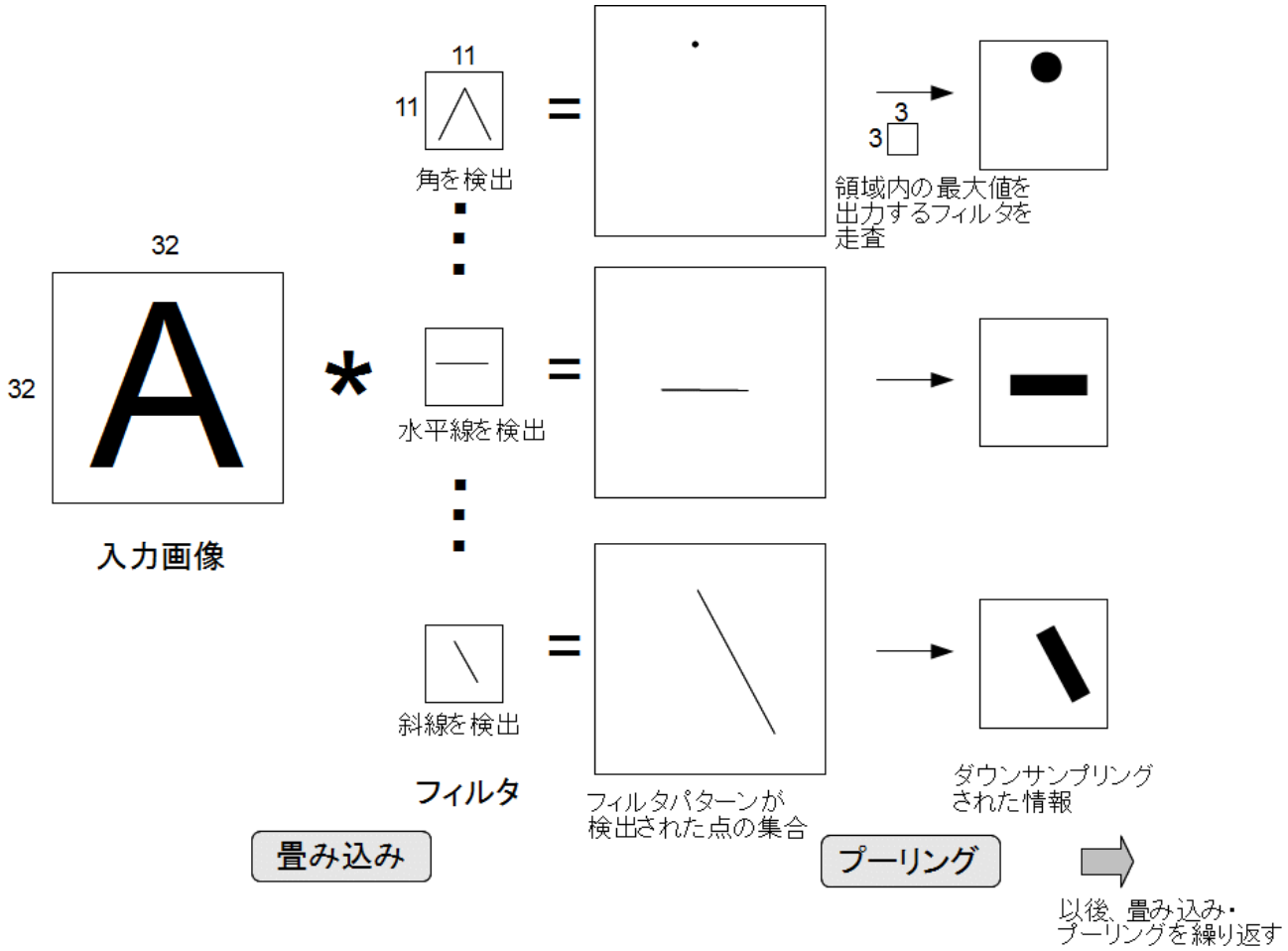
7.3.3 特化した構造をもつニューラルネットワーク

- 畳み込みニューラルネットワーク



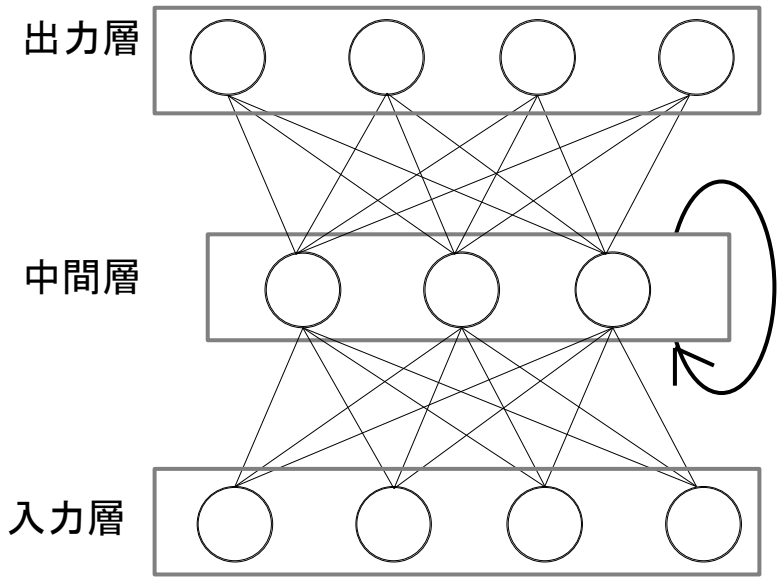
7.3.3 特化した構造をもつニューラルネットワーク

- 畳み込みニューラルネットワークの演算

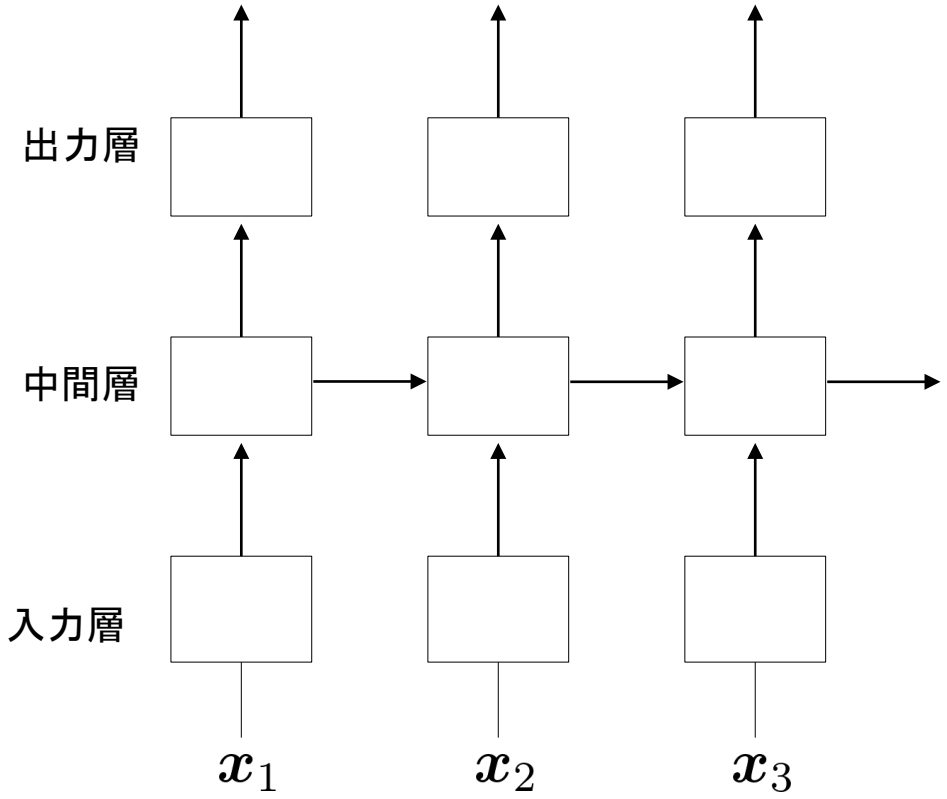


7.3.3 特化した構造をもつニューラルネットワーク

- 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



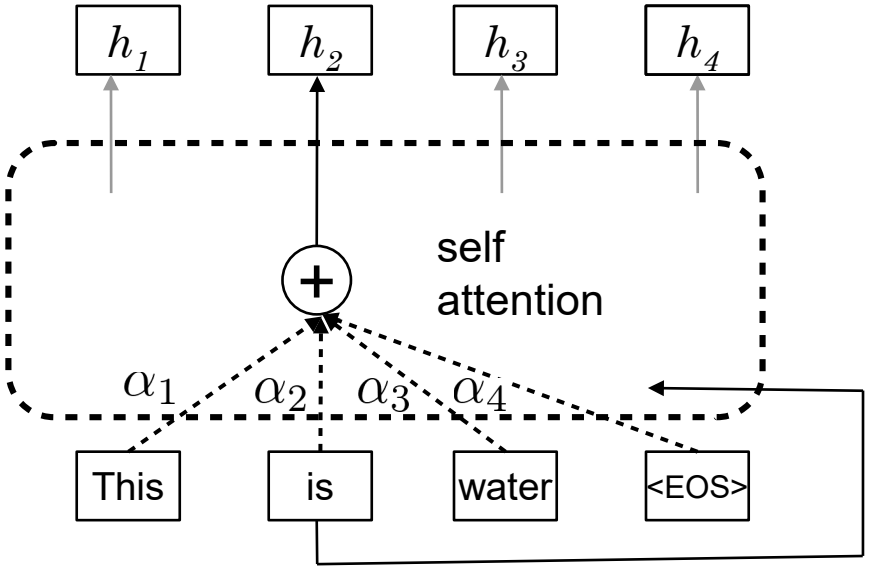
(a) リカレントニューラルネットワーク

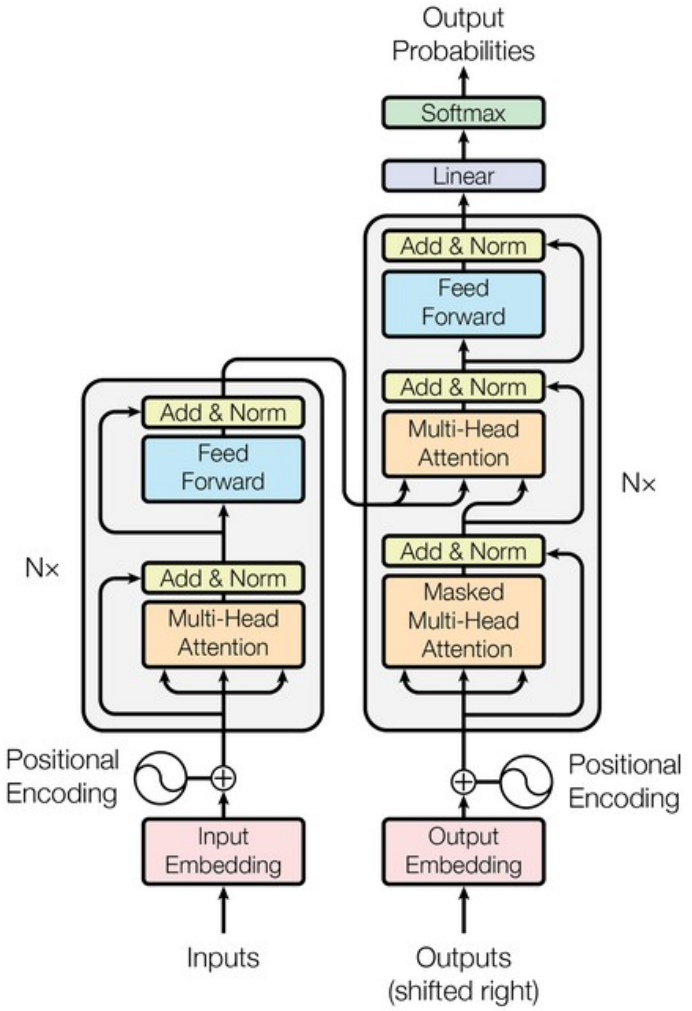


(b) 帰還路を時間方向に展開

7.3.3 特化した構造をもつニューラルネットワーク

- Transformer: Self-attention + フィードフォワードNN
  - 自分の中間表現を作るときに、入力の他の部分との関係を計算
  - BERTなどの事前学習モデルに使われる





Ashish Vaswani et.al. : Attention Is All You Need, arXiv:1706.03762

まとめ

- ニューラルネットワークは誤差を最小にする確率的勾配降下法の枠組みで非線形識別面を学習できる
- 多階層のニューラルネットワークは誤差逆伝播法を用いる
- 勾配消失問題などで学習がうまくゆかないことがあったが、現在では様々な工夫により深層学習が可能になっている