Chainer v2 による実践深層学習

第 2 章　ニューラルネットのおさらい

山木翔馬

2017/10/05

2.1 モデル

NN は m 次元のベクトル $\boldsymbol{x}$ を n 次元のベクトル $\boldsymbol{y}$ に射影する関数 f を推定する学習手法.

$$\boldsymbol{y} = f(\boldsymbol{x}) \qquad s.t \quad \boldsymbol{x} \in R^m,\ \boldsymbol{y} \in R^n$$

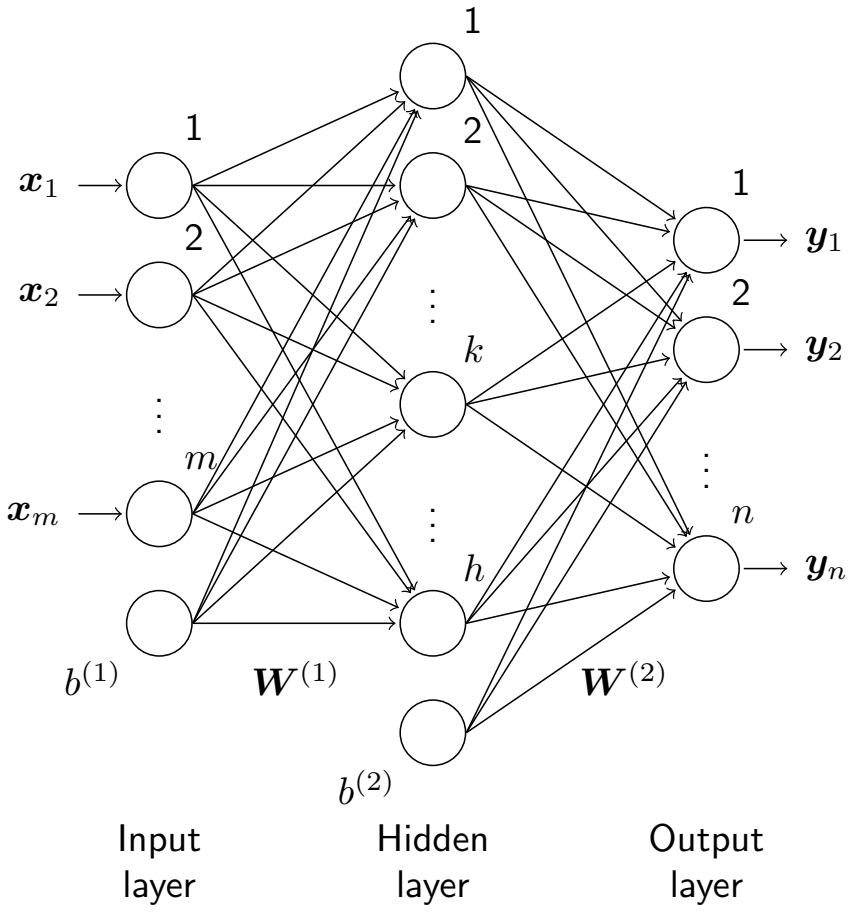
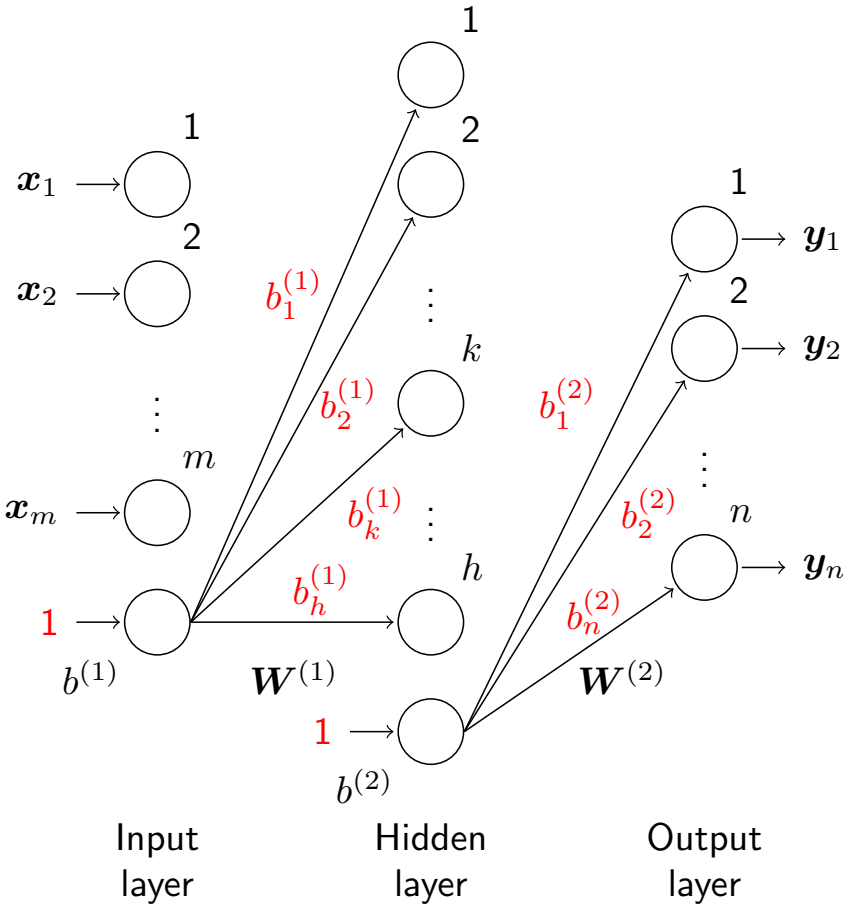
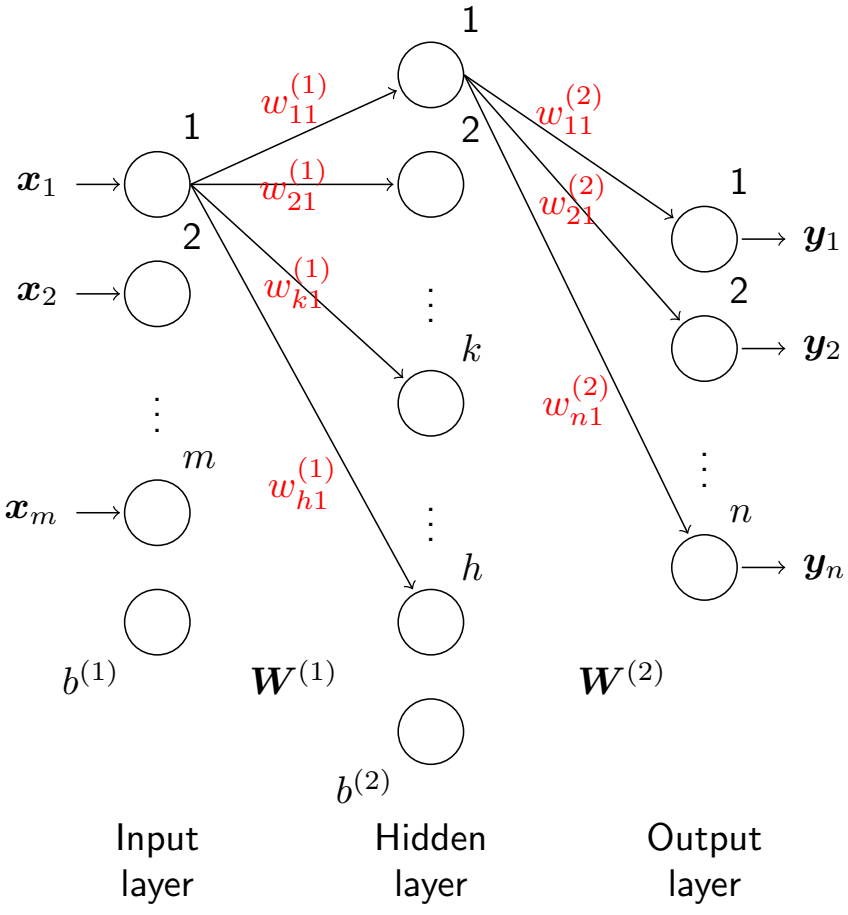


図 2. 1: NN による関数のモデル

- 入力層，隠れ層，出力層からなる
- 入力層のユニット m 個
- 中間層のユニット h 個
- 出力層のユニット n 個
- 層と層の間のエッジには重み $\boldsymbol{W}$
- 入力層と中間層にある $b^{(1)}, b^{(2)}$ はバイアスとよぶ



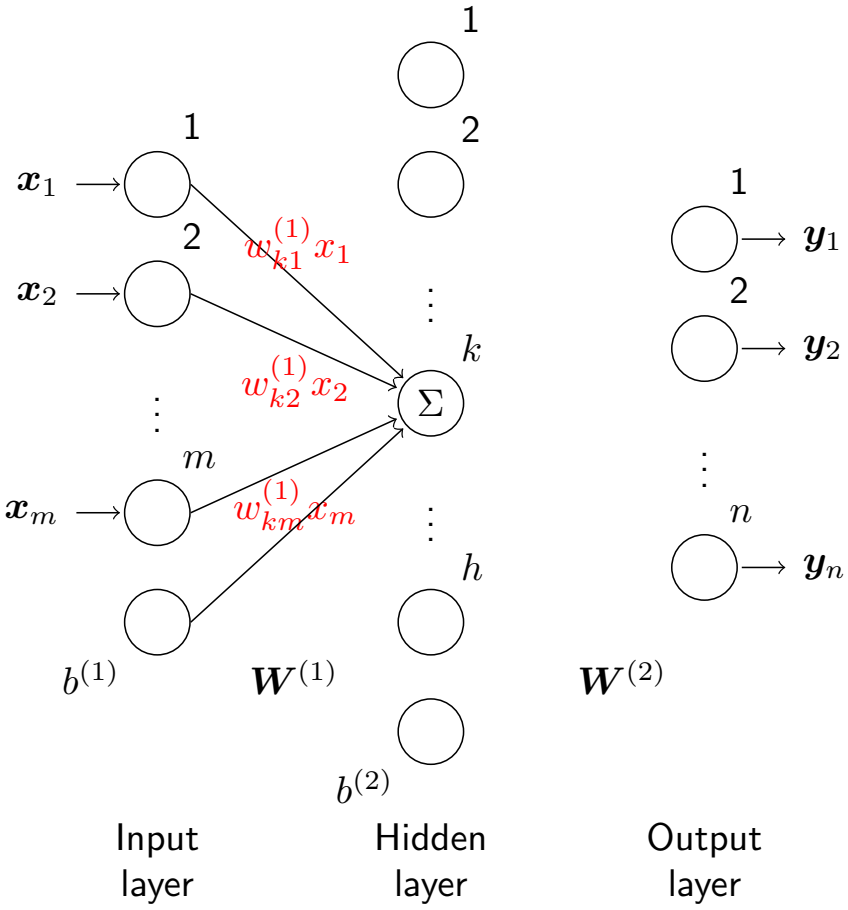
- バイアスのユニットからユニット i 間には $b_i^{(1)}, b_i^{(2)}$ という重みがある
- バイアスの出力はこの重みになる
  （ バイアスの入力が常に 1 で固定されているとも考えられる）



- 入力層のユニット i から中間層のユニット k には重み $w_{ki}^{(1)}$
- 中間層のユニット k から出力層のユニット j には重み $w_{jk}^{(2)}$

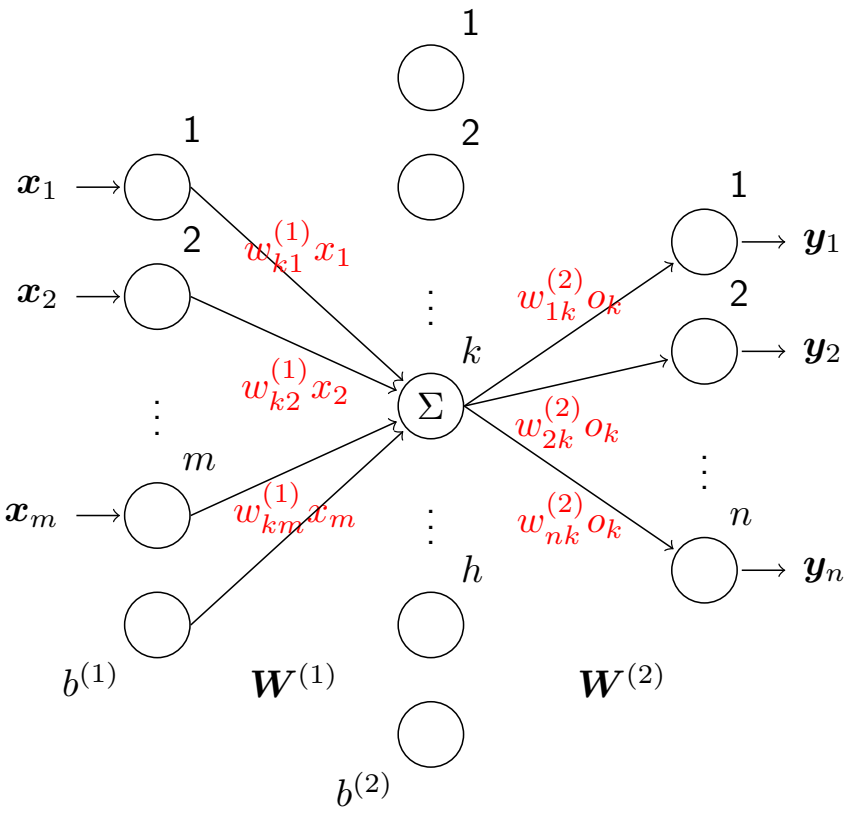
添字に注意!

a から b のとき w_{ba}



- 入力層のユニット i から中間層のユニット k への値は $w_{ki}^{(1)} x_i$
- 中間層のユニット k の入力は

$$\sum_{i=1}^{m} w_{ki}^{(1)} x_i + b_k^{(1)}$$



- 中間層のユニット k は入力値をある活性化関数 σ に与えた値 o_k を出力する

$$o_k = \sigma\left(\sum_{i=1}^{m} w^{(1)}_{ki} x_i + b^{(1)}_k\right)$$

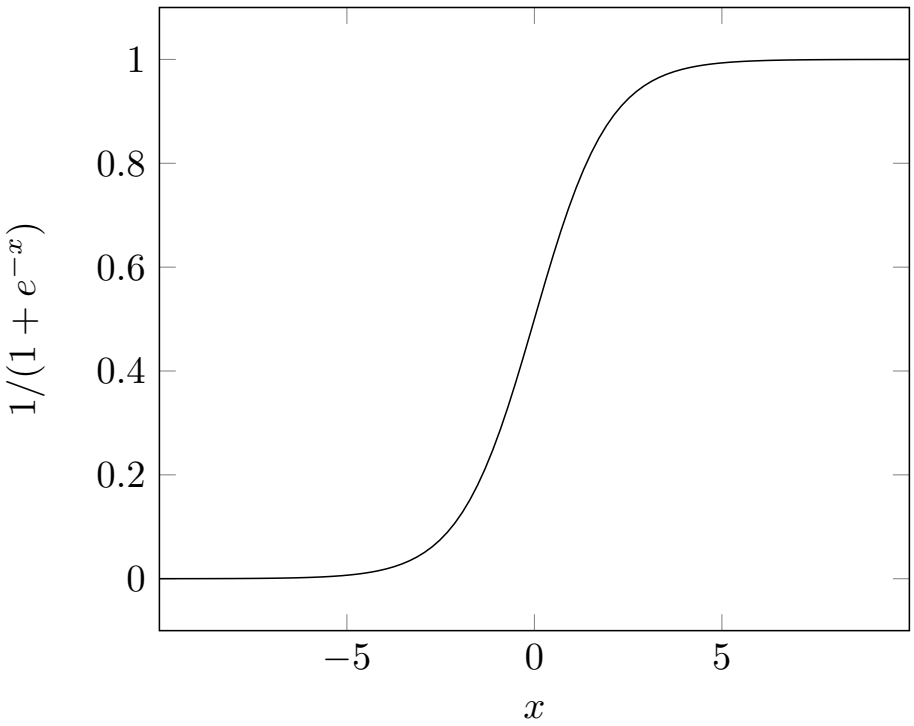
- 同様に，出力層のユニット j の入力は

$$\sum_{k=1}^{h} w^{(2)}_{jk} o_k + b^{(2)}_j$$

- 出力層のユニット j の出力は

$$\sigma\left(\sum_{k=1}^{h} w^{(2)}_{jk} o_k + b^{(2)}_j\right)$$

シグモイド 関数



シグモイド 関数

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

- 活性化関数としてよく使われる重要な関数
- 値域が 0 から 1 なので確率との相性が良い

入力層から中間層への重みを

$$\boldsymbol{W}^{(1)} = \begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \cdots & w_{1m}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & \cdots & w_{2m}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{h1}^{(1)} & w_{h2}^{(1)} & \cdots & w_{hm}^{(1)} \end{pmatrix}$$

とおき，中間層の出力を並べたものをベクトル $\boldsymbol{o}$ とおくと，

$$\boldsymbol{o} = \sigma_1\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right)$$

が成立する．同様に出力層の出力は

$$\sigma_2\left(\boldsymbol{W^{(2)}o} + \boldsymbol{b}^{(2)}\right) = \sigma_2\left(\boldsymbol{W^{(2)}}\sigma_1\left(\boldsymbol{W^{(1)}x} + \boldsymbol{b}^{(1)}\right) + \boldsymbol{b}^{(2)}\right)$$

結局，NN の関数 f のモデルは

$$\boldsymbol{y} = f(\boldsymbol{x}) = \sigma_2\left(\boldsymbol{W^{(2)}}\sigma_1\left(\boldsymbol{W^{(1)}x} + \boldsymbol{b}^{(1)}\right) + \boldsymbol{b}^{(2)}\right)$$

2.2 確率的勾配降下法と誤差逆伝搬法

$$\boldsymbol{y} = f(\boldsymbol{x}) = \sigma_2\left(\boldsymbol{W}^{(2)}\sigma_1\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right) + \boldsymbol{b}^{(2)}\right)$$

- 関数 f のモデルのパラメータ $\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}$（ 以下 $\boldsymbol{\theta}$ で表す）を訓練データから推定する.
- 適当なパラメータの初期値 $\boldsymbol{\theta}^{(0)}$ から始めて， $\boldsymbol{\theta}^{(i)}$ を $\theta^{(i+1)}$ に更新して $\boldsymbol{\theta}$ を求める.
- パラメータの更新には標準的に確率的勾配降下法(SGD) が使われる.

確率的勾配降下法(SGD)

訓練データ D の集合を

$$D = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \cdots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$$

とおく．SGD では訓練データの k 番目のデータ $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ の2 乗誤差

$$E_k = \frac{1}{2}|f(\boldsymbol{x}_k; \boldsymbol{\theta}^{(i)}) - \boldsymbol{y}_k|^2 = \frac{1}{2}\sum_{j=1}^{n}(f_j - y_j)^2$$

$$(f_j = f(\boldsymbol{x}_k; \boldsymbol{\theta}^{(i)}), \qquad y_j = \boldsymbol{y}_k\text{の } j \text{ 次元目の値})$$

を減少させるように $\theta^{(i)}$ を $\theta^{(i+1)}$ に更新.

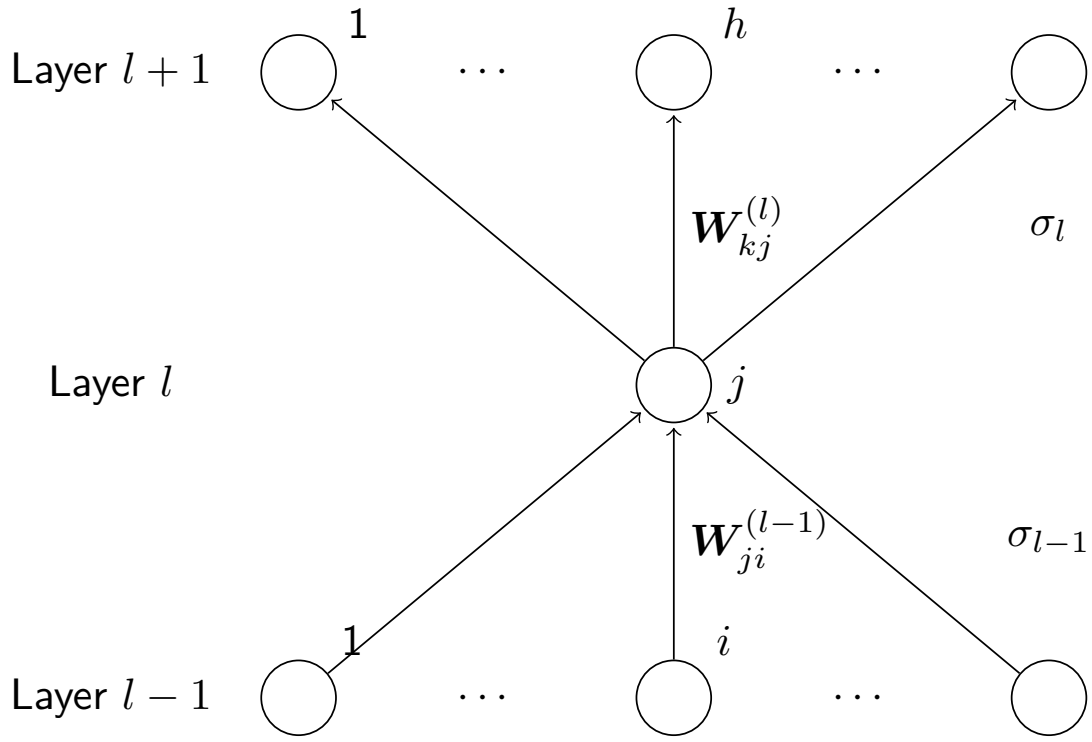
$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \alpha\Delta E_k \qquad (\alpha\text{は学習率})$$

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \alpha \Delta E_k$$

$$\Delta E_k = \left(\left. \frac{\partial E_k}{\partial \theta_1} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i)}}, \left. \frac{\partial E_k}{\partial \theta_2} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i)}}, \cdots, \left. \frac{\partial E_k}{\partial \theta_V} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(i)}} \right)$$

結局 SGD では各 $\frac{\partial E_k}{\partial \theta_i}$ が求まればよい.

どうやって求めるか?



第 l 層のユニット j の出力を作る際に，活性化関数に与える入力を $a_j^{(l)}$ とすると，第 l 層のユニット j の出力 $\sigma_l(a_j^{(l)})$ となる．

$$a_j^{(l)} = \sum_i w_{ji}^{(l-1)} \sigma_{l-1}(a_i^{(l-1)}) + b_j^{(l-1)}$$

合成関数の微分を使うと

$$\frac{\partial E_k}{\partial w_{ji}^{(l-1)}} = \frac{\partial E_k}{\partial a_j^{(l)}} \frac{\partial a_j^{(l)}}{\partial w_{ji}^{(l-1)}} = \frac{\partial E_k}{\partial a_j^{(l)}} \sigma_{l-1}(a_i^{(l-1)})$$

$$\frac{\partial E_k}{\partial b_j^{(l-1)}} = \frac{\partial E_k}{\partial a_j^{(l)}} \frac{\partial a_j^{(l)}}{\partial b_j^{(l-1)}} = \frac{\partial E_k}{\partial a_j^{(l)}}$$

が成り立つ.

つまり，第 $l-1$ 層と第 l 層の間に存在するパラメータは $\frac{\partial E_k}{\partial a_j^{(l)}}$ を計算すれば求められる.

さらに，多変数関数の合成関数の微分を使うと

$$\frac{\partial E_k}{\partial a_j^{(l)}} = \sum_h \frac{\partial E_k}{\partial a_h^{(l+1)}} \frac{\partial a_j^{(l)}}{\partial b_j^{(l-1)}} = \frac{\partial E_k}{\partial a_j^{(l)}}$$

が成り立ち，最終的に

$$\frac{\partial E_k}{\partial a_j^{(l)}} = \sum_h \frac{\partial E_k}{\partial a_h^{(l+1)}} w_{hj} \sigma_l^{'}(a_j^{(l)})$$

となる．つまり $\frac{\partial E_k}{\partial a_j^{(l)}}$ を計算するには，1 つ上の層の $\frac{\partial E_k}{\partial a_h^{(l+1)}}$ を計算できればよい．

出力層から入力層に向かって誤差を伝播させてパラメータを求めるため，誤差逆伝播法と呼ぶ．