ディープラーニングの仕組みを知ろう！

第 3 回　人工知能勉強会

Shion MORISHITA

July 18, 2024

目次

はじめに

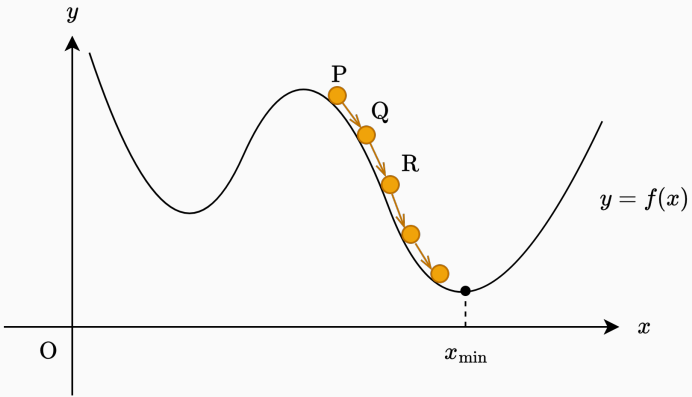
目的

-

勾配降下法の復習

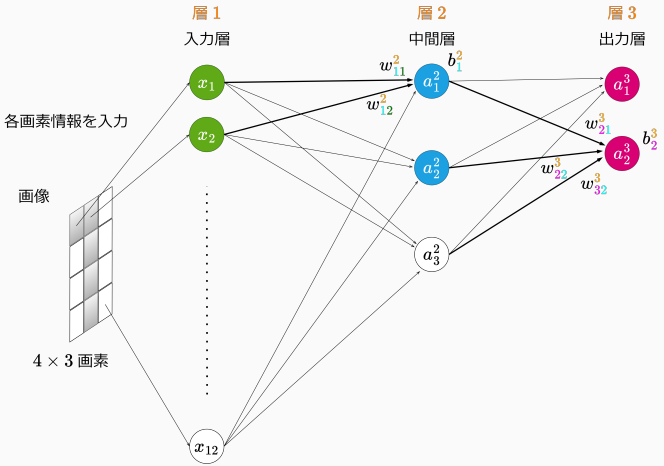
勾配降下法のイメージ

ボールが転がる方向に向かってパラメータ（ここでは x）を更新



ニューラルネットワークのパラメータ

$w^2_{11}, \ldots, w^3_{11}, \ldots, b^2_1, \ldots, b^3_1, \ldots$ **がパラメータ**



ニューラルネットワークへの勾配降下法の適用

$$\begin{bmatrix} \Delta w_{11}^2 \\ \vdots \\ \Delta w_{11}^3 \\ \vdots \\ \Delta b_1^2 \\ \vdots \\ \Delta b_1^3 \\ \vdots \end{bmatrix} = -\eta \begin{bmatrix} \dfrac{\partial C_{\mathrm{T}}}{\partial w_{11}^2} \\ \vdots \\ \dfrac{\partial C_{\mathrm{T}}}{\partial w_{11}^3} \\ \vdots \\ \dfrac{\partial C_{\mathrm{T}}}{\partial b_1^2} \\ \vdots \\ \dfrac{\partial C_{\mathrm{T}}}{\partial b_1^3} \\ \vdots \end{bmatrix}$$ を用いて、 $$\begin{bmatrix} w_{11}^2 \\ \vdots \\ w_{11}^3 \\ \vdots \\ b_1^2 \\ \vdots \\ b_1^3 \\ \vdots \end{bmatrix}$$ を $$\begin{bmatrix} w_{11}^2 + \Delta w_{11}^2 \\ \vdots \\ w_{11}^3 + \Delta w_{11}^3 \\ \vdots \\ b_1^2 + \Delta b_1^2 \\ \vdots \\ b_1^3 + \Delta b_1^3 \\ \vdots \end{bmatrix}$$ へ更新を繰り返す。

※正の小さな定数 η を学習係数といい、モデル作成者が自由に設定する。

勾配降下法の問題点

微分を実際に計算するのは大変

$$\frac{\partial C_{\mathrm{T}}}{\partial w_{11}^2} = \sum_{k=1}^{64} \frac{\partial C_k}{\partial w_{11}^2}$$

$$= \sum_{k=1}^{64} \left\{ \frac{\partial C_k}{\partial a_1^3[k]} \frac{\partial a_1^3[k]}{\partial z_1^3[k]} \frac{\partial z_1^3[k]}{\partial a_1^2[k]} \frac{\partial a_1^2[k]}{\partial z_1^2[k]} \frac{\partial z_1^2[k]}{\partial w_{11}^2} + \frac{\partial C_k}{\partial a_2^3[k]} \frac{\partial a_2^3[k]}{\partial z_2^3[k]} \frac{\partial z_2^3[k]}{\partial a_1^2[k]} \frac{\partial a_1^2[k]}{\partial z_1^2[k]} \frac{\partial a_1^2[k]}{\partial w_{11}^2} \right\}.$$

微分地獄の解決策

誤差逆伝播法 の導入

誤差逆伝播法

誤差逆伝播法

ユニットの誤差

誤差逆伝播法とは？

特徴

- 煩雑な微分計算を、数列の漸化式に置き換える
- ユニットの誤差（error）と呼ばれる変数 δ_j^l を用いる

ユニットの誤差 δ_j^l の導入

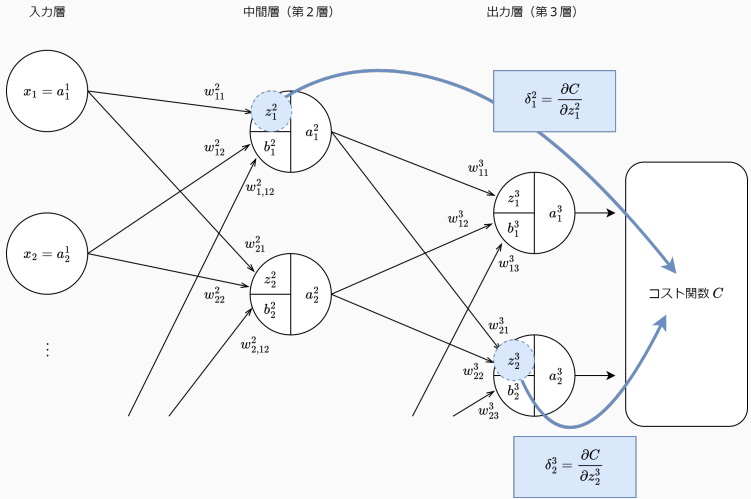
ユニットの誤差 δ_j^l を次のように定義する：

$$\delta_j^l \triangleq \frac{\partial C}{\partial z_j^l} \quad (l \geq 2). \tag{1}$$

なぜこれを導入したか？

- 微分の計算から漸化式の計算へ変えることができるから

ユニットの誤差 δ_j^l のイメージ



重みに関する 2 乗誤差の偏微分を δ_j^l で表現

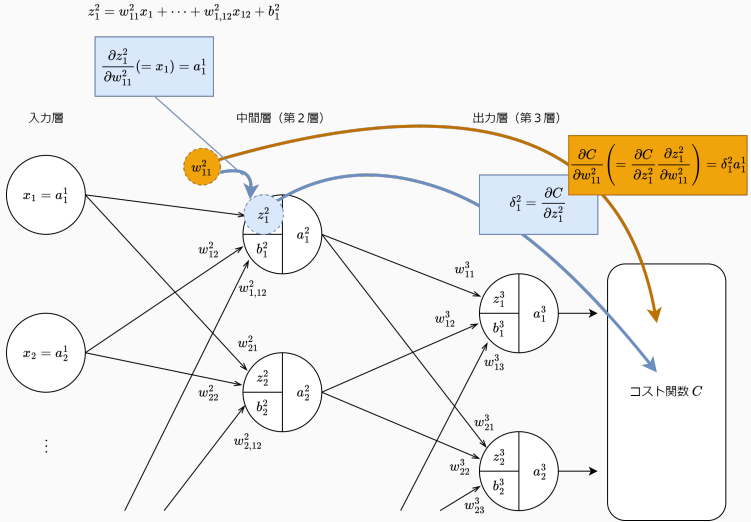
重みに関する偏微分をユニットの誤差 δ_j^l を用いて次にように表すことができる：

$$\frac{\partial C}{\partial w_{ji}^l} = \delta_j^l a_i^{l-1} \quad (l \geq 2). \tag{2}$$

ポイント

- δ_j^l の値さえ計算できれば、$\frac{\partial C}{\partial w_{ji}^l}$ を偏微分の計算なしで求めることができる！
  - a_i^{l-1} はユニットの出力値なので、普通に計算できる

重みに関する 2 乗誤差の偏微分のイメージ



バイアスに関する 2 乗誤差の偏微分を δ_j^l で表現

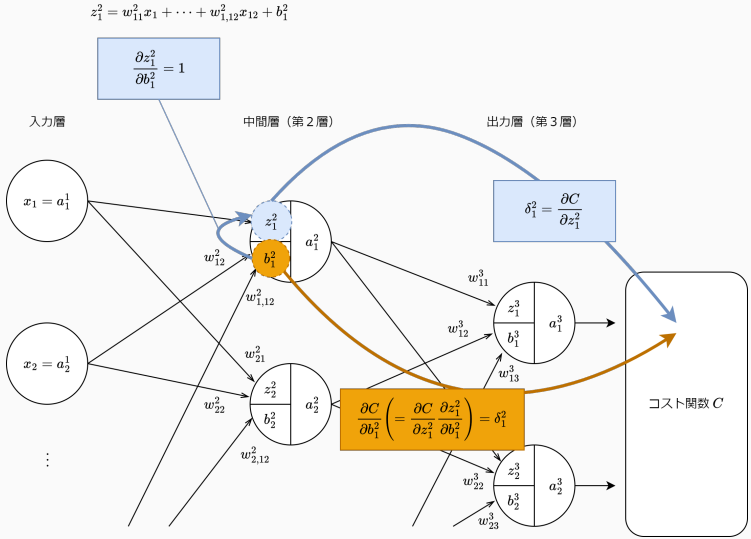
バイアスに関する偏微分をユニットの誤差 δ_j^l を用いて次にように表すことができる：

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \quad (l \geq 2). \tag{3}$$

ポイント

- δ_j^l の値さえ計算できれば、$\dfrac{\partial C}{\partial b_j^l}$ を偏微分の計算なしで求めることができる！

バイアスに関する２乗誤差の偏微分のイメージ



で、δ_j^l はどう計算するの？

ここでようやく漸化式の考え方が役立つ！

1. 出力層（第 L 層）のユニットの誤差 δ_j^L を計算する
2. l 層と $l+1$ 層のユニットの誤差の「（逆）漸化式」を用いて、出力層側のユニットの誤差から入力層側のユニットの誤差を計算する

つまり、δ_j^L が計算できれば、$\delta_j^{L-1}, \delta_j^{L-2}, \ldots, \delta_j^2$ とすべて計算できることになる

【重要】出力層の δ_j^L

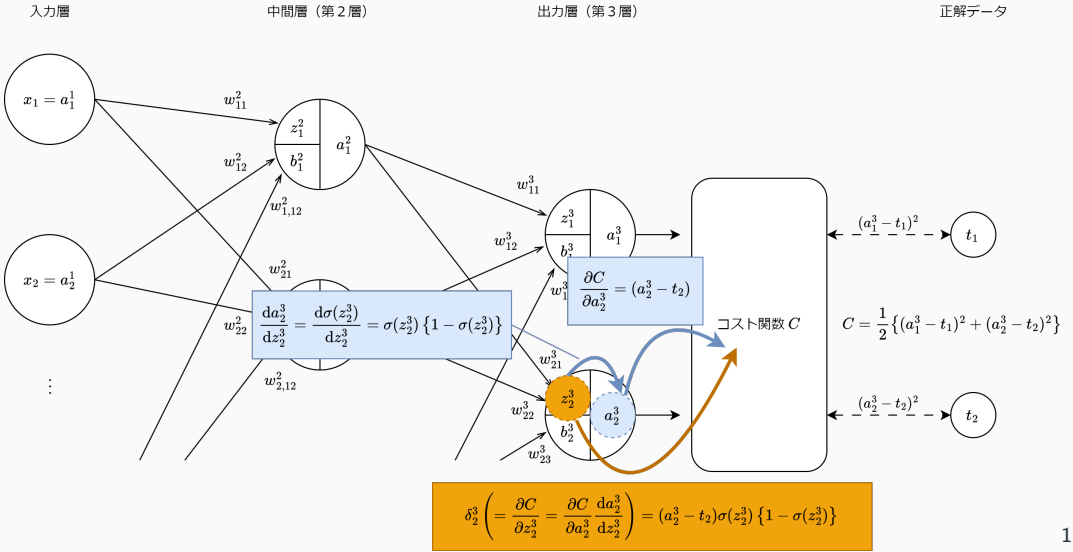
出力層のユニットの誤差 δ_j^L は次にように表すことができる：

$$\begin{aligned}\delta_j^L &= \frac{\partial C}{\partial a_j^L}\frac{\mathrm{d}a_j^L}{\mathrm{d}z_j^L} \\ &= (a_j^L - t_j)\cdot\sigma(z_j^L)\cdot\left\{1-\sigma(z_j^L)\right\}. \end{aligned} \tag{4}$$

ポイント

- $a_j^L,\ t_j,\ z_j^L$ はすべて分かっている！数値計算可能 !!

出力層の δ_j^L のイメージ

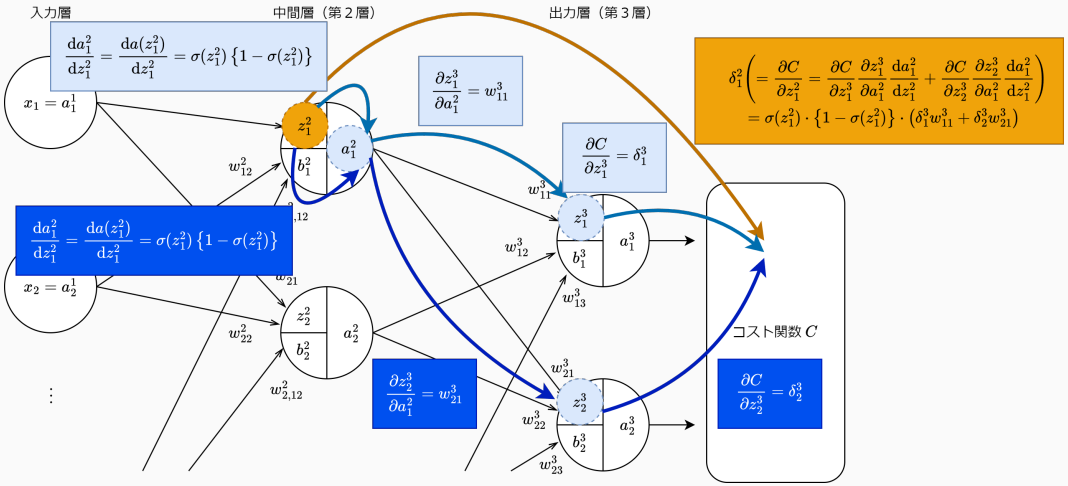


中間層の δ_j^l

中間層のユニットの誤差 δ_i^l は次のように表すことができる：

$$\begin{aligned}\delta_i^l &= \frac{\mathrm{d}a_i^l}{\mathrm{d}z_i^l} \sum_j \delta_j^{l+1} w_{ji}^{l+1} \\ &= \sigma(z_i^l) \cdot \left\{1 - \sigma(z_i^l)\right\} \cdot \sum_j \delta_j^{l+1} w_{ji}^{l+1} \quad (l \geq 2).\end{aligned} \tag{5}$$

中間層の δ_i^l のイメージ



中間層の δ_i^l の式をどう使う？

式 (5) で $l+1=L$ とすると、

$$\delta_i^{L-1} = \sigma(z_i^{L-1}) \left\{1 - \sigma(z_i^{L-1})\right\} \sum_j \delta_j^L w_{ji}^L.$$

- z_i^{L-1} はニューラルネットワークの重み付き入力なので、計算済み
- w_{ji}^L はニューラルネットワークの重みなので、ただの数字
- δ_j^L は出力層のユニットの誤差なので、計算可能（式 (4)）

δ_j^L が求まれば $\delta_i^{L-1}, \delta_i^{L-2}, \ldots, \delta_i^2$ とすべてのユニットの誤差 δ_i^l を求めることができる！

δ_i^l が計算できるようになると、、、？

- 式 (2), (3) を用いて $\dfrac{\partial C}{\partial w_{ji}^l}$ も $\dfrac{\partial C}{\partial b_j^l}$ も微分なしで計算できる！
- 勾配降下法の微分地獄を回避できた !!

誤差逆伝播法

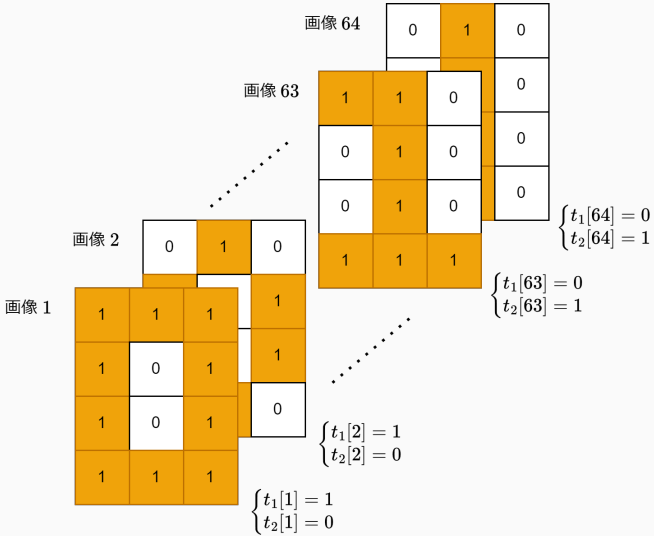
アルゴリズム

誤差逆伝播法のアルゴリズム

1. 学習データ・正解データの用意
2. 重み (w_{ji}^l)・バイアス (b_j^l) をランダムに、学習率 η を小さい正の値で設定
3. 重み付き入力 $(z_j^l[k])$、活性化関数の値 $(a_j^l[k])$、2 乗誤差 (C_k) を算出
4. 式 (4) を用いて出力層の「ユニットの誤差」$(\delta_j^L[k])$、式 (5) を用いて中間層の「ユニットの誤差」$(\delta_i^l[k])$ を算出
5. 式 (2), (3) を用いて、ユニットの誤差から 2 乗誤差 $\{C_k\}$ の偏微分 $\left(\dfrac{\partial C_k}{\partial w_{ji}^l}\right)$, $\left(\dfrac{\partial C_k}{\partial b_j^l}\right)$ を算出
6. 3.〜5. のデータを足し合わせることでコスト関数 C_{T} を算出し、その勾配成分 $(\Delta w_{ji}^l) = -\eta\left(\dfrac{\partial C_{\mathrm{T}}}{\partial w_{ji}^l}\right)$, $(\Delta b_j^l) = -\eta\left(\dfrac{\partial C_{\mathrm{T}}}{\partial b_j^l}\right)$ を算出[1]
7. 勾配降下法を利用して、6. で求めた勾配成分から重みとバイアスの値を更新
8. コスト関数 C_{T} が十分小さくなるまで、3.〜7. を反復

[1] この勾配成分をまとめたベクトル（勾配）を ∇C_{T} と表す。

1. 学習データ・正解データを用意



2. 重み、バイアス、学習率の初期設定

- $w^l_{ji} \longleftarrow$ random
- $b^l_j \longleftarrow$ random
- $\eta \longleftarrow$ random

3. ユニットの出力値、活性化関数の値、2 乗誤差の算出

- $z_j^l[k] \longleftarrow \sum_m w_{jm}^l a_m^{l-1}[k] + b_j^l, \quad a_j^1[k] = x_j[k]$
- $a_j^l[k] \longleftarrow \sigma(z_j^l[k]) = \dfrac{1}{1 + e^{-z_j^l[k]}}$
- $C_k \longleftarrow \dfrac{1}{2} \sum_m \left(t_m[k] - a_m^L[k]\right)^2$

4. 誤差逆伝播法から各層のユニットの誤差の算出

- $\delta_j^L[k] \longleftarrow (t_j[k] - a_j^L[k]) \cdot \sigma(z_j^L[k]) \cdot \left\{1 - \sigma(z_j^L[k])\right\}$
- $\delta_i^l[k] \longleftarrow \sigma(z_i^l[k]) \cdot \left\{1 - \sigma(z_i^l[k])\right\} \cdot \sum_j \delta_j^{l+1}[k] w_{ji}^{l+1}$

5. ユニットの誤差から 2 乗誤差の偏微分の算出

- $\dfrac{\partial C_{\mathrm{T}}}{\partial w_{ji}^{l}} \longleftarrow \sum_{k} \delta_{j}^{l}[k] a_{j}^{l-1}[k]$
- $\dfrac{\partial C_{\mathrm{T}}}{\partial b_{j}^{l}} \longleftarrow \sum_{k} \delta_{j}^{l}[k]$

6. コスト関数とその勾配成分の算出

- $C_{\mathrm{T}} \longleftarrow \sum_k C_k$
- $\Delta w_{ji}^l \longleftarrow -\eta \dfrac{\partial C_{\mathrm{T}}}{\partial w_{ji}^l}$
- $\Delta b_j^l \longleftarrow -\eta \dfrac{\partial C_{\mathrm{T}}}{\partial b_j^l}$

7. 重みとバイアスの更新

- $w^l_{ji} \longleftarrow w^l_{ji} + \Delta w^l_{ji}$
- $b^l_j \longleftarrow b^l_j + \Delta b^l_j$

誤差逆伝播法

数値判定 AI の実装

スプレッドシートでも AI が作れるよ

https://docs.google.com/spreadsheets/d/1ZquWGD6V3Q5JXeYjy7jB4zUDGW_Nuxvi5Aryu7YTJs4/edit?gid=1554072861#gid=1554072861