誤差逆伝播法：深層学習における重み更新のメカニズム

概要

本稿では、深層学習においてニューラルネットワークの重みを効率的に更新するための主要なアルゴリズムである誤差逆伝播法（Backpropagation）について、その数学的背景と具体的な導出を解説する。特に、連鎖律の多変数関数への応用を通じて、誤差がネットワークを逆方向に伝播していく過程を詳細に追う。

1　はじめに

深層学習の文脈において、ニューラルネットワークは多層の構造を持ち、その各層における重みパラメータを適切に調整することで、与えられたタスク（画像認識、自然言語処理など）の性能を向上させる。この重み調整のプロセスは「学習」と呼ばれ、その中心的なアルゴリズムが誤差逆伝播法である。

誤差逆伝播法は、ネットワークの出力と正解（教師データ）との間の誤差を計算し、その誤差をネットワークの入力層に向かって逆方向に伝播させることで、各重みが誤差にどれだけ寄与したかを効率的に評価し、更新することを可能にする。本稿では、この一連のプロセスを、特に多変数関数の連鎖律に焦点を当てて数学的に解説する。

2　ニューラルネットワークの基本構造と記号

ニューラルネットワークは、入力層、隠れ層（複数存在しうる）、出力層から構成される。各層のニューロンは、前の層からの入力に重みを乗じて合計し、バイアスを加えた後、活性化関数を通して次の層へ出力する。

2.1　記号の定義

- x_i: 入力層の i 番目のニューロンの値。
- $z_j^{(l)}$: l 層目の j 番目のニューロンへの活性化関数適用前の入力（重み付き線形和）。
- $y_j^{(l)}$: l 層目の j 番目のニューロンの活性化関数適用後の出力。
- d_j: 教師データ（正解）の j 番目の値。
- $w_{ij}^{(l)}$: $l-1$ 層目の i 番目のニューロンから l 層目の j 番目のニューロンへの重み。
- $b_j^{(l)}$: l 層目の j 番目のニューロンへのバイアス。
- E: 誤差関数。
- $f^{(l)}(u)$: l 層目の活性化関数。

2.2　ニューロンの計算

l 層目の j 番目のニューロンへの入力 $z_j^{(l)}$ は、以下のように計算される。

$$z_j^{(l)} = \sum_i w_{ij}^{(l)} y_i^{(l-1)} + b_j^{(l)}$$

そして、活性化関数を適用した後の出力 $y_j^{(l)}$ は以下のようになる。

$$y_j^{(l)} = f^{(l)}(z_j^{(l)})$$

2.3　主要な活性化関数とその導関数

実用上重要な活性化関数とその導関数を以下に示す：

1. **シグモイド関数**：
$$\sigma(x) = \frac{1}{1+e^{-x}}, \quad \sigma'(x) = \sigma(x)(1-\sigma(x))$$

2. **双曲線正接関数（tanh）**：
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad \tanh'(x) = 1 - \tanh^2(x)$$

3. **ReLU（Rectified Linear Unit）**：
$$\mathrm{ReLU}(x) = \max(0, x), \quad \mathrm{ReLU}'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

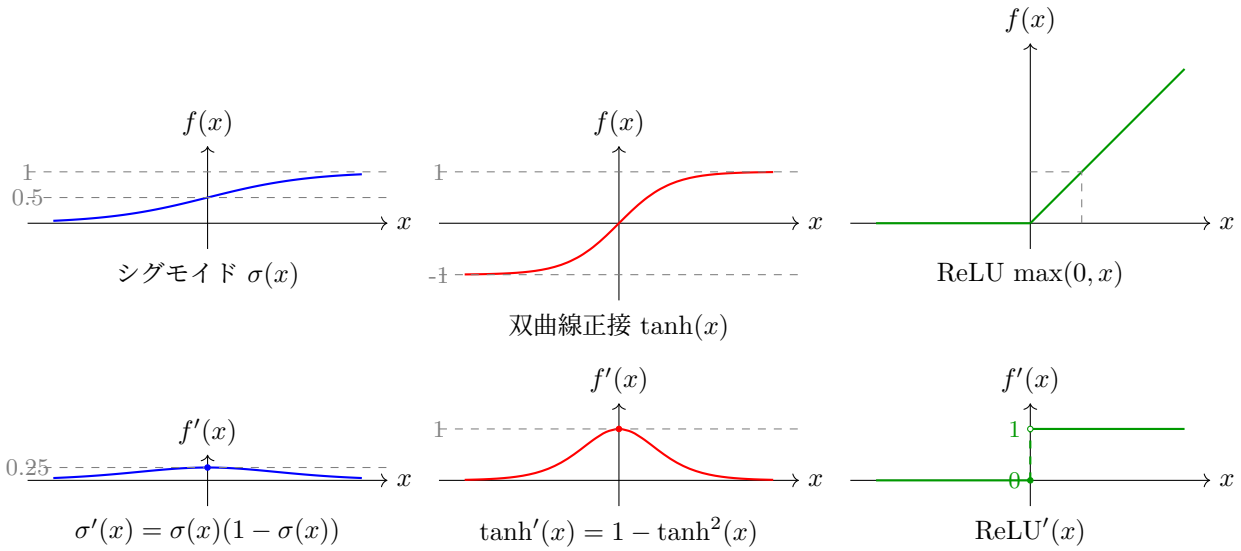

図 1　主要な活性化関数（上段）とその導関数（下段）。シグモイド関数の導関数は $x = 0$ で最大値 0.25 をとる一方、ReLU は正の入力に対して導関数が 1 で一定である。

3　誤差関数と重み更新の目標

学習の目標は、ネットワークの出力 $y_j^{(L)}$（L は出力層のインデックス）と教師データ d_j との間の誤差を最小化することである。ここでは、二乗誤差関数を例にとる。

$$E = \frac{1}{2}\sum_j (y_j^{(L)} - d_j)^2$$

重み更新は、この誤差関数 E を各重み $w_{ij}^{(l)}$ で偏微分し、その勾配に沿って重みを調整することで行われる。重みの更新式は以下の通りである。

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial E}{\partial w_{ij}^{(l)}}$$

ここで、η は学習係数である。

4　連鎖律の応用

誤差逆伝播法の核心は、多変数関数の連鎖律にある。E は $y^{(L)}$ の関数であり、$y^{(L)}$ は $z^{(L)}$ の関数、そして $z^{(L)}$ は $w^{(L)}$ の関数といったように、関数の連鎖として表現できるため、連鎖律を用いることで各重みに対する誤差の勾配を計算することができる。

4.1　出力層の重み更新の導出

出力層（$l = L$）における重み $w_{ij}^{(L)}$ に対する誤差 E の偏微分を考える。

$$\frac{\partial E}{\partial w_{ij}^{(L)}}$$

連鎖律を適用すると、これは次のように分解できる。

$$\frac{\partial E}{\partial w_{ij}^{(L)}} = \frac{\partial E}{\partial z_j^{(L)}} \frac{\partial z_j^{(L)}}{\partial w_{ij}^{(L)}}$$

ここで、それぞれの項を計算する。

4.1.1　$\frac{\partial z_j^{(L)}}{\partial w_{ij}^{(L)}}$ の計算

$z_j^{(L)} = \sum_k w_{kj}^{(L)} y_k^{(L-1)} + b_j^{(L)}$ であるため、$w_{ij}^{(L)}$ で偏微分すると、和の記号から $k = i$ の項のみが残り、係数として $y_i^{(L-1)}$ が得られる。

$$\frac{\partial z_j^{(L)}}{\partial w_{ij}^{(L)}} = y_i^{(L-1)}$$

4.1.2　$\frac{\partial E}{\partial z_j^{(L)}}$ の計算（誤差項 $\delta_j^{(L)}$）

この項を誤差項 $\delta_j^{(L)}$ と定義する。

$$\delta_j^{(L)} = \frac{\partial E}{\partial z_j^{(L)}}$$

さらに連鎖律を適用すると、

$$\delta_j^{(L)} = \frac{\partial E}{\partial y_j^{(L)}} \frac{\partial y_j^{(L)}}{\partial z_j^{(L)}}$$

ここで、出力層の活性化関数が恒等関数 $f^{(L)}(u) = u$ であると仮定すると、$y_j^{(L)} = z_j^{(L)}$ となるため、

$$\frac{\partial y_j^{(L)}}{\partial z_j^{(L)}} = \frac{\partial z_j^{(L)}}{\partial z_j^{(L)}} = 1$$

また、$E = \frac{1}{2}\sum_k (y_k^{(L)} - d_k)^2$ より、

$$\frac{\partial E}{\partial y_j^{(L)}} = \frac{\partial}{\partial y_j^{(L)}} \left(\frac{1}{2}(y_j^{(L)} - d_j)^2 \right) = (y_j^{(L)} - d_j)$$

したがって、出力層の誤差項は以下のようになる。

$$\delta_j^{(L)} = (y_j^{(L)} - d_j) \cdot 1 = y_j^{(L)} - d_j$$

まとめると、出力層の重み更新のための勾配は次のようになる。

$$\frac{\partial E}{\partial w_{ij}^{(L)}} = \delta_j^{(L)} y_i^{(L-1)} = (y_j^{(L)} - d_j) y_i^{(L-1)}$$

4.2 中間層の重み更新の導出

次に、中間層（$l < L$）における重み $w_{ij}^{(l)}$ に対する誤差 E の偏微分を考える。

$$\frac{\partial E}{\partial w_{ij}^{(l)}}$$

同様に連鎖律を適用すると、

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \frac{\partial E}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial w_{ij}^{(l)}}$$

4.2.1 $\frac{\partial z_j^{(l)}}{\partial w_{ij}^{(l)}}$ の計算

出力層の場合と同様に、

$$\frac{\partial z_j^{(l)}}{\partial w_{ij}^{(l)}} = y_i^{(l-1)}$$

4.2.2 $\frac{\partial E}{\partial z_j^{(l)}}$ の計算（誤差項 $\delta_j^{(l)}$）

この項も誤差項 $\delta_j^{(l)}$ と定義する。

$$\delta_j^{(l)} = \frac{\partial E}{\partial z_j^{(l)}}$$

ここで、この $z_j^{(l)}$ が次の層 $l+1$ の全てのニューロン $z_k^{(l+1)}$ に影響を与えることを考慮すると、多変数の連鎖律を用いる必要がある。

$$\delta_j^{(l)} = \sum_k \frac{\partial E}{\partial z_k^{(l+1)}} \frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}}$$

ここで、$\frac{\partial E}{\partial z_k^{(l+1)}}$ は次の層の誤差項 $\delta_k^{(l+1)}$ に他ならない。

$$\delta_j^{(l)} = \sum_k \delta_k^{(l+1)} \frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}}$$

さらに、$\frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}}$ を分解する。

$$\frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}} = \frac{\partial z_k^{(l+1)}}{\partial y_j^{(l)}} \frac{\partial y_j^{(l)}}{\partial z_j^{(l)}}$$

$z_k^{(l+1)} = \sum_m w_{mk}^{(l+1)} y_m^{(l)} + b_k^{(l+1)}$ であるため、

$$\frac{\partial z_k^{(l+1)}}{\partial y_j^{(l)}} = w_{jk}^{(l+1)}$$

また、$y_j^{(l)} = f^{(l)}(z_j^{(l)})$ であるため、

$$\frac{\partial y_j^{(l)}}{\partial z_j^{(l)}} = f^{(l)\prime}(z_j^{(l)})$$

したがって、中間層の誤差項は以下のようになる。

$$\delta_j^{(l)} = \left(\sum_k \delta_k^{(l+1)} w_{jk}^{(l+1)} \right) f^{(l)\prime}(z_j^{(l)})$$

この式は、次の層の誤差項と重みを用いて、現在の層の誤差項が計算できることを示している。これが「誤差の逆伝播」と呼ばれるゆえんである。

まとめると、中間層の重み更新のための勾配は次のようになる。

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \delta_j^{(l)} y_i^{(l-1)} = \left(\sum_k \delta_k^{(l+1)} w_{jk}^{(l+1)} \right) f^{(l)\prime}(z_j^{(l)}) y_i^{(l-1)}$$

4.3 バイアス項の更新

バイアス項 $b_j^{(l)}$ に対する誤差の勾配は以下のように導出される：

$$\frac{\partial E}{\partial b_j^{(l)}} = \frac{\partial E}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial b_j^{(l)}} = \delta_j^{(l)} \cdot 1 = \delta_j^{(l)}$$

したがって、バイアス項の更新式は：

$$b_j^{(l)} \leftarrow b_j^{(l)} - \eta \delta_j^{(l)}$$

5 計算効率性の分析

誤差逆伝播法の重要性は、その計算効率性にある。N 個のパラメータを持つネットワークにおいて：

- **数値微分法**：各パラメータの勾配を計算するために $O(N)$ 回の順伝播が必要
- **誤差逆伝播法**：1 回の順伝播と 1 回の逆伝播で全パラメータの勾配を計算（$O(1)$ 回）

この効率性により、大規模ネットワークの学習が現実的になる。

6　結論

誤差逆伝播法は、ニューラルネットワークの学習において不可欠なアルゴリズムである。出力層から誤差項を計算し、それを連鎖律を用いて一つ前の層へと順に伝播させることで、各層の重みに対する誤差の勾配を効率的に計算することができる。これにより、勾配降下法などの最適化アルゴリズムを用いて、ネットワークの重みを効果的に更新し、モデルの性能を向上させることが可能となる。

本稿で示した導出は、活性化関数が恒等関数の場合を含む基本的な二乗誤差関数に基づいているが、他の誤差関数や活性化関数に対しても、連鎖律を適切に適用することで同様の原理で誤差逆伝播法を適用できる。

7　練習問題

7.1　問題 1：シンプルな 2 層ネットワークでの手計算

以下の 2 層ニューラルネットワークで誤差逆伝播法を手計算。

ネットワーク構造：

- 入力層：1 ニューロン (x)
- 隠れ層：2 ニューロン (h_1, h_2)
- 出力層：1 ニューロン (y)
- 活性化関数：隠れ層は ReLU、出力層は恒等関数

初期値：

- 入力：$x = 1$
- 重み：$w_{11}^{(1)} = 0.5$（$x \to h_1$）、$w_{12}^{(1)} = -0.3$（$x \to h_2$）、$w_{11}^{(2)} = 0.8$（$h_1 \to y$）、$w_{21}^{(2)} = 0.4$（$h_2 \to y$）
- バイアス：全て 0
- 教師データ：$d = 0.2$
- 学習率：$\eta = 0.1$

(a) 順伝播を計算し、出力 y を求めよ。
(b) 誤差 $E = \frac{1}{2}(y - d)^2$ を計算せよ。
(c) 出力層の誤差項 $\delta^{(2)}$ を計算せよ。
(d) 隠れ層の誤差項 $\delta_1^{(1)}, \delta_2^{(1)}$ を計算せよ。
(e) 各重みの勾配 $\frac{\partial E}{\partial w}$ を計算せよ。
(f) 更新後の重みを計算せよ。

7.2　問題 2：XOR 問題における誤差逆伝播法

XOR（排他的論理和）は線形分離不可能な問題として知られている。

XOR の真理値表：

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

ネットワーク構造：

- 入力層：2 ニューロン
- 隠れ層：2 ニューロン（シグモイド活性化）
- 出力層：1 ニューロン（シグモイド活性化）

(a) 単層パーセプトロン（隠れ層なし）で XOR 問題が解けない理由を図示して説明せよ。

(b) 以下の初期重みから始めて、入力 $(1, 0) \to$ 出力 1 の学習を 1 ステップ実行せよ：

- $w_{11}^{(1)} = 0.5, w_{21}^{(1)} = 0.5$
- $w_{12}^{(1)} = 0.5, w_{22}^{(1)} = 0.5$
- $w_{11}^{(2)} = 0.5, w_{21}^{(2)} = 0.5$

(c) 誤差逆伝播法なしで（出力層の重みのみを更新して）同じ問題を解こうとした場合、なぜ学習が困難になるか説明せよ。