誤差逆伝播法（II）

誤差逆伝播法 (I) では、簡単なニューラルネットワークを例にその考え方を説明した。ここでは、一般的なネットワークに適用できるように誤差逆伝播法の定式化を行う。

1. 多層ニューラルネットワーク

まず、図 1 に示す多層ニューラルネットワークを例として、定式化に用いる記号と各層の入出力関係について説明する。

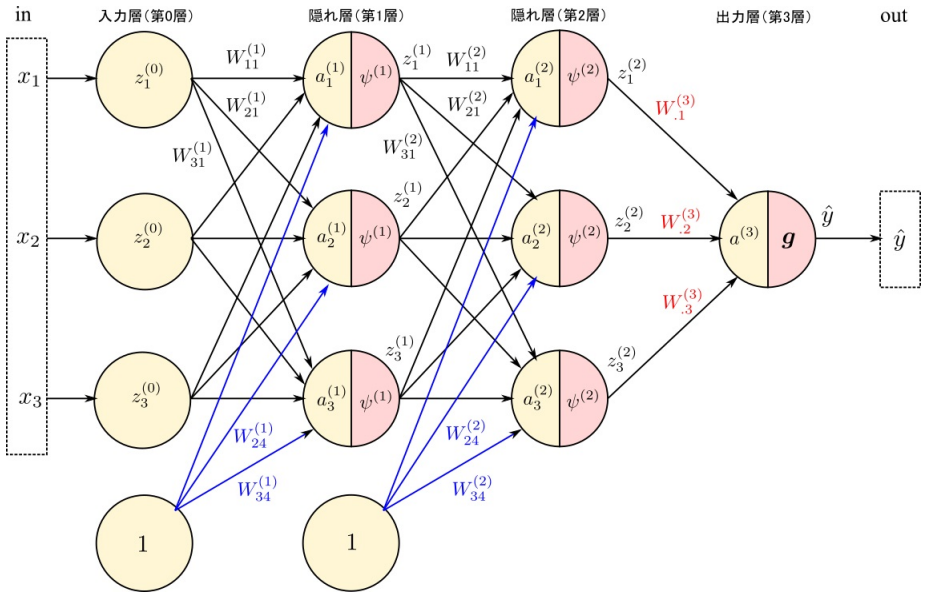

図 1　多層ネットワークの例。入力層は第 0 層であり、その値はベクトル $\boldsymbol{x}$ であるが、これは $\boldsymbol{z}^{(0)}$ でもある。第 1 層への重み行列 $\boldsymbol{W}^{(1)}$ と第 2 層への重み行列 $\boldsymbol{W}^{(2)}$ は 3 行 4 列の行列で表される。$W_{\mu\nu}^{(1)}$ は第 0 層の ν 番目のユニットから第 1 層の μ 番目のユニットへの結合に対する重みである。第 3 層への重み行列 $\boldsymbol{W}^{(3)}$ の列数は 3 であるが、行数は出力関数の引数の次元に依存する。隠れユニットでは活性 $\boldsymbol{a}^{(l)} = \boldsymbol{W}^{(l)}\boldsymbol{z}^{(l-1)}$ から活性化関数 $\psi^{(l)}$ によって $z_\mu = \psi^{(l)}(a_\mu^{(l)})$ が計算される。出力ユニットでは活性 $\boldsymbol{a}^{(3)} = \boldsymbol{W}^{(3)}\boldsymbol{z}^{(2)}$ が出力関数 $\boldsymbol{g}$ によって変換される。$\hat{\boldsymbol{y}} = \boldsymbol{g}(\boldsymbol{a}^{(3)})$ がネットワークの最終的な出力である。

入力データ（特徴量ベクトル）は

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

である。これを入力層ベクトルとして

$$\boldsymbol{z}^{(0)} = \begin{pmatrix} z_1^{(0)} \\ z_2^{(0)} \\ z_3^{(0)} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

で表す。

$\hat{\boldsymbol{y}}$ がネットワークの最終的な出力（予測値）であり、出力関数 $\boldsymbol{g}$ によって計算される。出力関数の例はシグモイド関数やソフトマックス関数である。ソフトマックス関数の出力はベクトルであるが、シグモイド関数の出力はスカラーであるので、それを 1 次元のベクトルであると捉え、出力関数を太文字の $\boldsymbol{g}$ で表す。

入力層と出力層の間にあるのが隠れ層である。ネットワーク全体で $K+1$ 層あるとすると、入力層が第 0 層、出力層が第 K 層、それ以外は隠れ層（$K-1$ 層存在）である。図 1 の場合は、全体で 4 層あ

り、入力層は第 0 層、出力層は第 3 層であり、隠れ層は第 1 層、第 2 層であり、隠れ層が 2 層存在するネットワークである。

層の間ではそれぞれの層のユニットの組み合わせの数だけ結合が存在する。たとえば第 l 層に m 個、第 $l-1$ 層に $n-1$ ユニットが存在するとき、それらの組み合わせの数である $m \times (n-1)$ 種類の結合が存在し、また第 l 層のそれぞれのユニットにバイアス項が必要になるため、あわせて $m \times n$ 個の重みが必要である。これを m 行 n 列の行列で表す。ただし重み行列は層ごとに異なるため、第 l 層への入力を計算するために使う重み行列を $\boldsymbol{W}^{(l)}$ で表す。すなわち

$$\boldsymbol{W}^{(l)} = \begin{pmatrix} W_{11}^{(l)} & W_{12}^{(l)} & \cdots & W_{1n}^{(l)} \\ W_{21}^{(l)} & W_{22}^{(l)} & \cdots & W_{2n} \\ \vdots & \vdots & & \vdots \\ W_{m1}^{(l)} & W_{m2}^{(l)} & \cdots & W_{mn}^{(l)} \end{pmatrix}$$

であり、第 $l-1$ 層の ν 番目のユニットから第 l 層の μ 番目のユニットへの結合の重みが行列の成分 $W_{\mu\nu}^{(l)}$ で表される：

$$W_{\mu\nu}^{(l)} \quad \begin{array}{l} l \ : \text{レイヤー番号} \\ \mu \ : \text{出力先となるユニットの番号} \\ \nu \ : \text{入力元となるユニットの番号} \end{array} \tag{1}$$

また、第 l 層の μ 番目のユニットへのバイアス項は $W_{\mu n}^{(l)}$ で表される。全体で $K+1$ 層ある場合、入力層に続く重み行列が $\boldsymbol{W}^{(1)}$、出力層の手前の重み行列が $\boldsymbol{W}^{(K)}$ である。

第 $l-1$ 層の各ユニットの出力を並べた隠れ層の出力ベクトルを $\boldsymbol{z}^{(l-1)}$ で表す：

$$\boldsymbol{z}^{(l-1)} = \begin{pmatrix} z_1^{(l-1)} \\ z_2^{(l-1)} \\ \vdots \\ z_{n-1}^{(l-1)} \end{pmatrix} \tag{2}$$

このとき第 l 層の μ 番目のユニットが受け取る活性は

$$a_\mu^{(l)} = \sum_{\nu=1}^{n} W_{\mu\nu}^{(l)} z_\nu^{(l-1)} = \left(\boldsymbol{W}^{(l)} \cdot \boldsymbol{z}^{(l-1)} \right)_\mu \tag{3}$$

である。活性を第 l 層のすべてのユニットについて並べて得られるのが活性ベクトルであり

$$\boldsymbol{a}^{(l)} = \boldsymbol{W}^{(l)} \cdot \boldsymbol{z}^{(l-1)} = \begin{pmatrix} \left(\boldsymbol{W}^{(l)} \cdot \boldsymbol{z}^{(l-1)} \right)_1 \\ \left(\boldsymbol{W}^{(l)} \cdot \boldsymbol{z}^{(l-1)} \right)_2 \\ \vdots \\ \left(\boldsymbol{W}^{(l)} \cdot \boldsymbol{z}^{(l-1)} \right)_m \end{pmatrix} \tag{4}$$

と表せる。

活性ベクトル $\boldsymbol{a}^{(l)}$ の各成分を活性化関数 $\psi^{(l)}$ で変換したものが第 l 層の出力となる。たとえば第 l 層の μ 番目のユニットの出力は $\psi^{(l)}(a_\mu^{(l)})$ であり、これを $z_\mu^{(l)}$ で表す:

$$z_\mu^{(l)} = \psi^{(l)}(a_\mu^{(l)}) \tag{5}$$

以下では、活性化関数 $\psi^{(l)}$ の集まりをベクトル値関数 $\boldsymbol{f}^{(l)}$ で表す。第 l 層の出力ベクトル $\boldsymbol{z}^{(l)}$ は $\boldsymbol{f}^{(l)}$ を使って以下のように計算できる。ただしバイアス項を表現するため、$\boldsymbol{f}$ の最後の成分を 1 にしている。また、$\boldsymbol{z}^{(l)}$ の次元を n で表している：

$$\boldsymbol{z}^{(l)} = \boldsymbol{f}^{(l)}(\boldsymbol{a}^{(l)}) = \begin{pmatrix} \psi^{(l)}(a_1^{(l)}) \\ \psi^{(l)}(a_2^{(l)}) \\ \vdots \\ \psi^{(l)}(a_{n-1}^{(l)}) \\ 1 \end{pmatrix} \tag{6}$$

活性の計算 $\boldsymbol{a}^{(l+1)} = \boldsymbol{W}^{(l+1)} \cdot \boldsymbol{z}^{(l)}$ において、$\boldsymbol{W}^{(l+1)}$ の最右列は $\boldsymbol{z}^{(l)}$ の最後の成分である 1 と掛けられるため、第 $l+1$ 層の各ユニットに対するバイアス項を表わすことになる。実際 $W_{\xi n}^{(l+1)}$ は $z_n^{(l)} = 1$ と掛けられるため、第 l 層のユニットの出力には依存せず、第 $l+1$ 層の ξ 番目のユニットに対するバイアス項となる。

以上をまとめると、多層ニューラルネットワークで行われる計算は以下のようにまとめられる ：

$$
\begin{aligned}
\boldsymbol{z}^{(0)} &= \boldsymbol{x} \\
\boldsymbol{a}^{(l)} &= \boldsymbol{W}^{(l)} \boldsymbol{z}^{(l-1)} \quad (1 \leq l \leq K) \\
\boldsymbol{z}^{(l)} &= \boldsymbol{f}^{(l)}(\boldsymbol{a}^{(l)}) \quad (1 \leq l \leq K) \\
\hat{\boldsymbol{y}} &= \boldsymbol{g}(\boldsymbol{a}^{(K)})
\end{aligned}
\tag{7}
$$

隠れ層の出力 $\boldsymbol{z}^{(l)}$ はサンプルを分類する上での中間表現とみなせる。入力の特徴量ベクトル $\boldsymbol{x}$ を隠れ層の間の重み行列による変換を通し、潜在的特徴量を並べたベクトルに変換することで、直線や平面を使った分類が行えるようになるというのが深層学習の仕組みである。

機械学習では観測できる確率変数を $\boldsymbol{x}$、観測できない潜在的な確率変数を $\boldsymbol{z}$ で表すことが多い。隠れ層の出力を $\boldsymbol{z}^{(l)}$ で表すのはそこから来ている。なお、以下ではすべての l について $\boldsymbol{z}^{(l)}$ を「隠れ層の出力」と表現する。厳密に言えば $\boldsymbol{z}^{(0)}$ はニューラルネットワークへの入力 $\boldsymbol{x}$ であるが、隠れ層の出力と言ったときには $\boldsymbol{z}^{(0)}$ も含むことにする。

なお、線形分離器を重ねるだけでは線形関数（1 次式）しか表せない。つまり線形分離器を複数重ねても、それは結局ひとつの線形分離器で表せてしまう（線形変換を合成しても線形変換にしかならない）。しかし、多層ニューラルネットワークでは層の間に活性化関数が挟まっていることにより、非線形の関数も表せる。表現力が高いのである。

2. 深層学習におけるパラメータベクトル

ニューラルネットワークにおいては、パラメータはユニット間の結合の重みである。階層的なニューラルネットワークでは重み行列 $\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}, \cdots, \boldsymbol{W}^{(K)}$ と表されるが、これらの行列の成分をすべて 1 列に並べたものをベクトル $\boldsymbol{\theta}$ で表す：

$$
\boldsymbol{\theta} = \begin{pmatrix} W_{11}^{(1)} \\ \vdots \\ W_{11}^{(2)} \\ \vdots \\ W_{11}^{(K)} \\ \vdots \end{pmatrix}
\tag{8}
$$

すると、損失関数は $J(\boldsymbol{\theta})$ と書ける。目標は $J(\boldsymbol{\theta})$ を最小化する $\boldsymbol{\theta}$ を見つけることである。そのような値を $\boldsymbol{\theta}^*$ と書くことにする。

深層学習ではパラメータベクトル $\boldsymbol{\theta}$ を繰り返し更新することで $\boldsymbol{\theta}^*$ に近づけていくというアプローチが取られる。これは反復的アルゴリズムの一種である。まずはパラメータの初期値 $\boldsymbol{\theta}^{(0)}$ を決め、それをデータに基づいて少しづつ更新することを繰り返す。そしてパラメータの更新にともなう損失関数の変化があらかじめ定められた目標値以下になったら学習を終了する。

3. 勾配降下法

まずはパラメータ更新を議論するための数式による記法を導入する。パラメータベクトル $\boldsymbol{\theta}$ と同じ次元を持つパラメータ空間を考える。$\boldsymbol{\theta}$ はパラメータ空間における位置を表わしている。パラメータ更新における各ステップは t で表す。t は 0 から始まる整数である。ステップ t におけるパラメータの値を $\boldsymbol{\theta}^{(t)}$ で表すと、$\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \cdots$ という系列はパラメータ空間における移動の過程と捉えられる。移

動方向を表すベクトルを $\Delta\boldsymbol{\theta}^{(t)}$ とすると、パラメータ更新操作は以下のように表せる。

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \Delta\boldsymbol{\theta}^{(t)} \tag{9}$$

$J(\boldsymbol{\theta})$ を最小化する $\boldsymbol{\theta}$ を見つけるためには、パラメータ空間において $J(\boldsymbol{\theta})$ の変化量がもっとも大ききなる方向と逆向きに動かせばよい。$J(\boldsymbol{\theta})$ の変化量がもっとも大ききなる方向は、勾配ベクトル

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = (\nabla_{\theta_1} J(\boldsymbol{\theta}), \nabla_{\theta_2} J(\boldsymbol{\theta}), \cdots) \tag{10}$$

によって与えられる。ここで、勾配ベクトルは各変数での偏微分を並べて得られる行ベクトルとして考える（勾配ベクトルを行ベクトルではなく列ベクトルとして表すこともできるが、後述のヤコビアンを定義する際に便利なため、勾配ベクトルを行ベクトルとして定義する）。

機械学習の目的は損失関数を減らすことであるので、勾配と逆方向の $-\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ という方向にパラメータ $\boldsymbol{\theta}^{(t)}$ を更新する。以下のようにパラメータ更新のベクトル $\Delta\boldsymbol{\theta}^{(t)}$ を決めることを**勾配降下法**（gradient descent）と呼ぶ。

$$\Delta\boldsymbol{\theta}^{(t)} = -\eta^{(t)} \left(\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})\right)^T \tag{11}$$

ここで、$\eta^{(t)}$ は**学習率**（learning rate）と呼ばれる値であり、パラメータ更新の大きさを決めている。また、式 (10) において勾配ベクトルを行ベクトルと定義しているので、式 (11) では転置することで列ベクトルとしている。

最も急な方向に動くことで降下していくので、勾配降下法は**最急降下法**（steepest descent）とも呼ばれる。勾配 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ が 0 である地点に到達したときには $\Delta\boldsymbol{\theta}^{(t)} = 0$ となるので、それ以上パラメータが更新されなくなるため、そこで学習が終了する。

以上より、勾配降下法による各ステップにおけるパラメータ更新は以下のように行われる ：

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta^{(t)} \left(\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})\right)^T \tag{12}$$

4. 確率的勾配降下法

深層学習では、すべての訓練データを使った経験リスクの最小化ではなく、訓練データ集合から繰り返しサンプルを取り出し、それを使って勾配を計算するアプローチが取られる。このアプローチを**確率的勾配降下法**（stochastic gradient descent：SGD）と呼ぶ。

訓練データ集合から抽出されたサンプルの集合を**ミニバッチ**（mini-batch）あるいは単に**バッチ**（batch）と呼ぶ。データの一部のみを抜き出すことで、確率的なサンプリングを行っているとみなせる。

パラメータ空間における損失関数には、一般に

- **大域的最適解**（global optimum）：パラメータ空間全体の中で損失関数が最小になる点
- **局所最適解**（local optimum）：局所的に最小となる点
- **プラトー**（plateau）：勾配の小さな領域が大きく広がった領域
- **鞍点**（saddle point）：ある次元については極小値、別の次元については極大値となる点

が存在している。

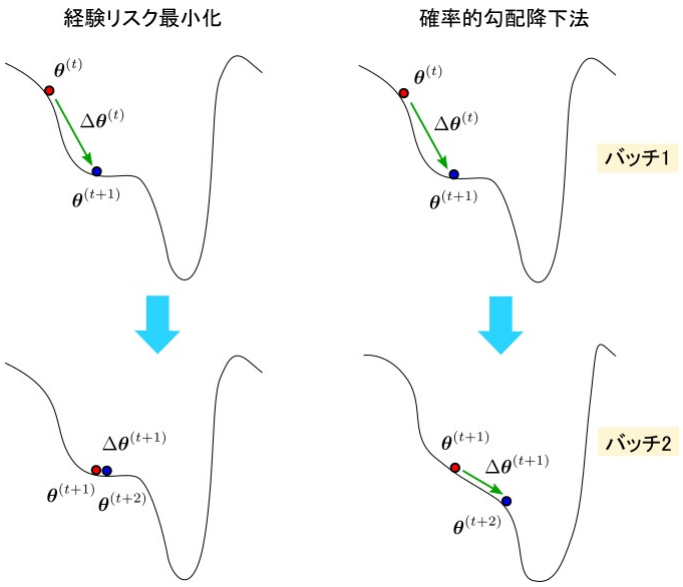

図 2　バッチの変化によるプラトーからの脱出。経験リスク最小化の場合、パラメータベクトル $\boldsymbol{\theta}$ がプラトーに入ると、勾配が小さいため、そこからなかなか抜け出せない。確率的勾配降下法ではバッチごとに損失関数の形状が変化するためプラトーから抜け出せる可能性が高い。

経験リスク最小化では、全空間での損失関数が用いられるためパラメータベクトルが局所最適解やプラトー、鞍点に入ってしまうと、勾配が小さい、あるいは 0 になるため、パラメータの更新が進まなくなる。これに対して確率的勾配降下法ではバッチごとに損失関数が変化するので、そのような状況から抜け出すことができる。

確率的勾配降下法のもう一つの利点はバッチごとに計算を行うので、すべてのデータを保存する必要がないことである。また、過去に利用したバッチをふたたび参照する必要もない。そのため大規模なデータ保存装置が不要になる。このようにデータを一括して使用せずに済むプログラムをオンラインアルゴリズム（online algorithm）と呼ぶ。確率的勾配降下法はオンラインアルゴリズムの例である。

5.　誤差逆伝播法（バックプロパゲーション）

勾配降下法では勾配 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ を使ってパラメータの更新が行われる。順伝播型ニューラルネットワーク（feedforward neural network）において、$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ を効率的に（無駄なく）求める工夫が誤差逆伝播法（backpropagation）である。誤差逆伝播法は以下の特徴を持つ。

- 順伝播と逆伝播：パラメータの現在の値を使って活性や隠れユニットの出力を計算する順伝播と、それに基づいてパラメータを更新する逆伝播を交互に繰り返す。
- ヤコビ行列の連鎖律による勾配の定義：ヤコビ行列の連鎖律を使うことでパラメータ更新に使用する勾配が再帰的に定義される。
- デルタの伝播：デルタの伝播という形で勾配を効率的に計算できる。

図 1 のような多層ニューラルネットワークは順伝播型ニューラルネットワークの一種であるが、$\boldsymbol{\theta}$ は式 (8) のように各層の重み行列の成分を並べて作られる。$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ はきわめて次元の高いベクトルであるが、そのうち、第 l 層の直前の重み行列 $\boldsymbol{W}^{(l)}$ についての微分を並べて得られる部分を $\nabla_{\boldsymbol{W}^{(l)}} J$ で表す。$\boldsymbol{W}^{(l)}$ はベクトルでなくて行列であるが、その各成分を一列に並べてベクトルにし、それで J を微分することで得られる勾配を $\nabla_{\boldsymbol{W}^{(l)}} J$ で表す。たとえば $\boldsymbol{W}^{(l)}$ が 2×2 行列であるとき、$\nabla_{\boldsymbol{W}^{(l)}} J$ は以下のようになる。

$$\nabla_{\boldsymbol{W}^{(l)}} J = \left(\frac{\partial J}{\partial W_{11}^{(l)}}, \frac{\partial J}{\partial W_{12}^{(l)}}, \frac{\partial J}{\partial W_{21}^{(l)}}, \frac{\partial J}{\partial W_{22}^{(l)}} \right) \tag{13}$$

総数が $K+1$ のとき、l が 1 から K のそれぞれについて $\nabla_{\boldsymbol{W}^{(l)}} J$ を求めれば $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ が得られる。

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = (\nabla_{\boldsymbol{W}^{(1)}} J, \nabla_{\boldsymbol{W}^{(2)}} J, \cdots, \nabla_{\boldsymbol{W}^{(K)}} J) \tag{14}$$

勾配 $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ が求まれば、式 (12) によりパラメータベクトル $\boldsymbol{\theta}$ を更新できる。

誤差逆伝播法では $\nabla_{\boldsymbol{W}^{(K)}} J$ をまず求め、続いて $\nabla_{\boldsymbol{W}^{(K-1)}} J$、さらに $\nabla_{\boldsymbol{W}^{(K-2)}} J$ と進めていき、最後に $\nabla_{\boldsymbol{W}^{(1)}} J$ を計算する。出力側から入力側に進むので、「逆」伝播という名前がついている。この順序で計算を行うことで計算コストを減らすことができる。

5.1 順伝播での予測

順伝播ではパラメータの現在の値 $\boldsymbol{\theta}^{(t)}$ を使い、訓練データについて予測を行う。確率的勾配降下法に基づく誤差逆伝播法では訓練データの一部に対してのみ予測が行われる。具体的には訓練データから $\tilde{n}$ 個のサンプルからなるバッチを取り出し、その中にあるサンプルを入力 $\boldsymbol{x}^{(i)}$ として、パラメータ $\boldsymbol{\theta}^{(t)}$ を持つニューラルネットワークに入力し、出力 $\hat{\boldsymbol{y}}^{(i)}$ を得る。この計算を行う際、**最終的な出力 $\hat{\boldsymbol{y}}^{(i)}$ だけでなく、各隠れ層の出力（中間表現）$\boldsymbol{z}^{(l,i)}$ や活性 $\boldsymbol{a}^{(l,i)}$ もすべて保存しておく**。

各バッチの中でそれぞれに属すサンプル i についてそれぞれ順伝播が行われ $\hat{\boldsymbol{y}}^{(i)}, \cdots, \hat{\boldsymbol{y}}^{(\tilde{n})}$ が計算される。バッチ全体について順伝播を行ったのち、逆伝播に移る。

5.2 出力ユニット直前の重み行列の更新規則

逆伝播では、もっとも出力に近い重み行列 $\boldsymbol{W}^{(K)}$ による勾配 $\nabla_{\boldsymbol{W}^{(K)}} J$ の計算から始める。$\boldsymbol{W}^{(K)}$ は出力関数がシグモイド関数の場合は行ベクトル、ソフトマックス関数の場合は行列である。

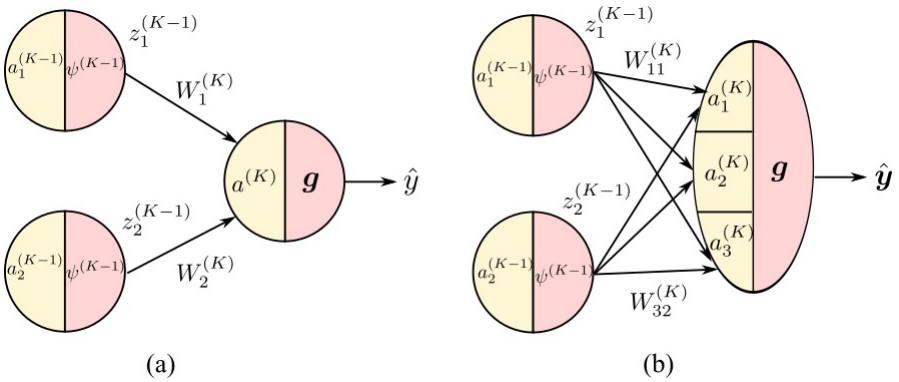

図 3　出力ユニット直前の重み行列と出力関数 $\boldsymbol{g}$　(a) $\boldsymbol{g}$：シグモイド関数の場合　(b) $\boldsymbol{g}$：ソフトマックス関数の場合

たとえば図 3(a) の 2 値分類でシグモイド関数を用いる場合は

$$\boldsymbol{W}^{(K)} = \left(W_1^{(K)}, W_2^{(K)}\right)$$

であり、活性は

$$a^{(K)} = \boldsymbol{W}^{(K)} \cdot \boldsymbol{z}^{(K-1)} = \left(W_1^{(K)}, W_2^{(K)}\right) \begin{pmatrix} z_1^{(K-1)} \\ z_2^{(K-1)} \end{pmatrix} = W_1^{(K)} z_1^{(K-1)} + W_2^{(K)} z_2^{(K-1)}$$

とスカラーとなる。この値がシグモイド関数の引数となる。一方、図 3(b) の 3 値分類でソフトマックス関数を用いる場合は

$$\boldsymbol{W}^{(K)} = \begin{pmatrix} W_{11}^{(K)} & W_{12}^{(K)} \\ W_{21}^{(K)} & W_{22}^{(K)} \\ W_{31}^{(K)} & W_{32}^{(K)} \end{pmatrix}$$

であり、活性は

$$\boldsymbol{a}^{(K)} = \boldsymbol{W}^{(K)} \cdot \boldsymbol{z}^{(K-1)} = \begin{pmatrix} W_{11}^{(K)} & W_{12}^{(K)} \\ W_{21}^{(K)} & W_{22}^{(K)} \\ W_{31}^{(K)} & W_{32}^{(K)} \end{pmatrix} \begin{pmatrix} z_1^{(K-1)} \\ z_2^{(K-1)} \end{pmatrix} = \begin{pmatrix} W_{11}^{(K)} z_1^{(K-1)} + W_{12}^{(K)} z_2^{(K-1)} \\ W_{21}^{(K)} z_1^{(K-1)} + W_{22}^{(K)} z_2^{(K-1)} \\ W_{31}^{(K)} z_1^{(K-1)} + W_{32}^{(K)} z_2^{(K-1)} \end{pmatrix}$$

と 3 次元のベクトルとなる。このベクトルがソフトマックス関数の引数となる。

一般に、第 $K-1$ 層のユニット数を m、ソフトマックス関数の引数の次元を n とすれば、第 K 層（出力層）の活性は

$$\boldsymbol{a}^{(K)} = \boldsymbol{W}^{(K)} \cdot \boldsymbol{z}^{(K-1)} = \begin{pmatrix} \sum_{\gamma=1}^{m} W_{1\gamma}^{(K)} z_\gamma^{(K-1)} \\ \vdots \\ \sum_{\gamma=1}^{m} W_{n\gamma}^{(K)} z_\gamma^{(K-1)} \end{pmatrix} \tag{15}$$

で与えられる n 次元ベクトルとなる。

以下では、出力関数としてソフトマックス関数、損失関数としてクロスエントロピの期待値を使う場合について考えていく。

「損失関数とクロスエントロピー」の式 (28) で示したように、単層ネットワークの出力層でソフトマックス関数を使った場合、入出力関係を表す確率分布 $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{W})$ は以下のように与えられる。

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{W}) = \frac{\exp\left(\boldsymbol{y}^T \boldsymbol{a}\right)}{\sum_k \exp\left(a_k\right)} = \frac{\exp\left(\boldsymbol{y}^T \boldsymbol{W}\boldsymbol{x}\right)}{\sum_k \exp\left((\boldsymbol{W}\boldsymbol{x})_k\right)} \tag{16}$$

これに対し、多層ニューラルネットワークの場合、出力ユニットに渡されるのは入力 $\boldsymbol{x}$ ではなく、直前の隠れ層の出力 $\boldsymbol{z}^{(K-1)}$ である。またそれに掛けられる重み行列は $\boldsymbol{W}^{(K)}$ であり、活性は $\boldsymbol{a}^{(K)} = \boldsymbol{W}^{(K)}\boldsymbol{z}^{(K-1)}$ である。このことから、式 (16) に対応した表式は

$$p(\boldsymbol{y}|\boldsymbol{a}^{(K)}) = \frac{\exp\left(\boldsymbol{y}^T \boldsymbol{a}^{(K)}\right)}{\sum_k \exp\left(a_k^{(K)}\right)} \tag{17}$$

となる。

損失関数としてクロスエントロピーの期待値を使う場合、その値は

$$E_{q(\boldsymbol{x}, \boldsymbol{y})}\left[-\log p\left(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}\right)\right]$$

である。条件部に入っているのは $\boldsymbol{x}$ と $\boldsymbol{\theta}$ であるが、出力は最終的には（出力ユニットに渡される活性）$\boldsymbol{a}^{(K)}$ によって決まるので、それを条件とすることもできる。すなわち

$$E_{q(\boldsymbol{x}, \boldsymbol{y})}\left[-\log p\left(\boldsymbol{y}|\boldsymbol{a}^{(K)}\right)\right]$$

と表せる。これに式 (17) を代入すると、以下が得られる。

$$\begin{aligned} J &= E_{q(\boldsymbol{x}, \boldsymbol{y})}\left[-\log p\left(\boldsymbol{y}|\boldsymbol{a}^{(K)}\right)\right] \\ &= E_{q(\boldsymbol{x}, \boldsymbol{y})}\left[-\log\left(\frac{\exp\left(\boldsymbol{y}^T \boldsymbol{a}^{(K)}\right)}{\sum_k \exp\left(a_k^{(K)}\right)}\right)\right] \\ &= E_{q(\boldsymbol{x}, \boldsymbol{y})}\left[-\boldsymbol{y}^T \boldsymbol{a}^{(K)} + \log\left(\sum_k \exp\left(a_k^{(K)}\right)\right)\right] \end{aligned}$$

損失関数 J はデータの分布 $q(\boldsymbol{x}, \boldsymbol{y})$ についての期待値であるので、期待値計算の対象となる関数を L で表す。すなわち J と L は以下の関係にある。

$$J(\boldsymbol{\theta}) = E_{q(\boldsymbol{x}, \boldsymbol{y})}\left[L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})\right] \tag{18}$$

すなわち J はパラメータ $\boldsymbol{\theta}$ のみの関数であるのに対し、L はデータの値 $\boldsymbol{x}$ と $\boldsymbol{y}$ にも依存する。今回の場合、以下が言える。ただし、活性 $\boldsymbol{a}^{(K)}$ は入力 $\boldsymbol{x}$ とパラメータ $\boldsymbol{\theta}$ の関数である。

$$L(\boldsymbol{x},\, \boldsymbol{y},\, \boldsymbol{\theta}) = -\log p\left(\boldsymbol{y}|\boldsymbol{a}^{(K)}\right) = -\boldsymbol{y}^T\boldsymbol{a}^{(K)} + \log\left(\sum_k \exp\left(a_k^{(K)}\right)\right) \tag{19}$$

この節では $\partial L/\partial W_{\mu\nu}^{(K)}$ を計算することを目標とする。ここで $W_{\mu\nu}^{(K)}$ は第 $K-1$ 層の ν 番目のユニットからソフトマックス関数の引数 $\boldsymbol{a}^{(K)}$ の第 μ 成分への結合に対する重みである。$\partial L/\partial W_{\mu\nu}^{(K)}$ の期待値が $\partial J/\partial W_{\mu\nu}^{(K)}$ になり、これが勾配降下法による J の最小化に使用できる。

微分の連鎖律により

$$\frac{\partial L}{\partial W_{\mu\nu}^{(K)}} = \frac{\partial L}{\partial \boldsymbol{a}^{(K)}}\frac{\partial \boldsymbol{a}^{(K)}}{\partial W_{\mu\nu}^{(K)}} \tag{20}$$

と表される。ここで、式 (19) と次元 r のソフトマックス関数 $\boldsymbol{S}(\boldsymbol{a}^{(K)})$ が

$$\boldsymbol{S}(\boldsymbol{a}^{(K)}) = \begin{pmatrix} S_1(\boldsymbol{a}^{(K)}) \\ S_2(\boldsymbol{a}^{(K)}) \\ \vdots \\ S_r(\boldsymbol{a}^{(K)}) \end{pmatrix} = \frac{1}{\sum_k^r \exp\left(a_k^{(K)}\right)} \begin{pmatrix} \exp\left(a_1^{(K)}\right) \\ \exp\left(a_2^{(K)}\right) \\ \vdots \\ \exp\left(a_r^{(K)}\right) \end{pmatrix} \tag{21}$$

で与えられることに注意すれば

$$\begin{aligned} \frac{\partial L}{\partial a_\lambda^{(K)}} &= -y_\lambda + \frac{\exp\left(a_\lambda^{(K)}\right)}{\sum_k \exp\left(a_k^{(K)}\right)} \\ &= -y_\lambda + S_\lambda(\boldsymbol{a}^{(K)}) = S_\lambda(\boldsymbol{a}^{(K)}) - y_\lambda \end{aligned} \tag{22}$$

となる。ベクトルの形にまとめると

$$\nabla_{\boldsymbol{a}^{(K)}} L = \frac{\partial L}{\partial \boldsymbol{a}^{(K)}} = \left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)^T \tag{23}$$

と表される。ここで、$\nabla_{\boldsymbol{a}^{(K)}} L$ はスカラー L を列ベクトル $\boldsymbol{a}^{(K)}$ 微分したものであるので L の勾配である。したかって、r 次元の行ベクトルとして定義するため右辺では転置を行っている。

次に、式 (15) より $a_\lambda^{(K)} = \sum_\gamma W_{\lambda\gamma}^{(K)} z_\gamma^{(K-1)}$ であることから

$$\frac{\partial a_\lambda^{(K)}}{\partial W_{\mu\nu}^{(K)}} = \frac{\partial}{\partial W_{\mu\nu}^{(K)}}\left(\sum_\gamma W_{\lambda\gamma}^{(K)} z_\gamma^{(K-1)}\right) = \delta_{\mu\lambda} z_\nu^{(K-1)} \tag{24}$$

となる。これは r 次元の列ベクトルで

$$\frac{\partial \boldsymbol{a}^{(K)}}{\partial W_{\mu\nu}^{(K)}} = \begin{pmatrix} \delta_{\mu 1} z_\nu^{(K-1)} \\ \delta_{\mu 2} z_\nu^{(K-1)} \\ \vdots \\ \delta_{\mu r} z_\nu^{(K-1)} \end{pmatrix}$$

である。

勾配 $\partial L/\partial \boldsymbol{a}^{(K)}$ は r 次元の行ベクトル、$\partial \boldsymbol{a}^{(K)}/\partial W_{\mu\nu}^{(K)}$ は r 次元の列ベクトルであるので、そのスカラー積として式 (20) が求まる。式 (22)、(24) を式 (20) へ代入して

$$\begin{aligned} \frac{\partial L}{\partial W_{\mu\nu}^{(K)}} &= \sum_\lambda \frac{\partial L}{\partial a_\lambda^{(K)}}\frac{\partial a_\lambda^{(K)}}{\partial W_{\mu\nu}^{(K)}} \\ &= \sum_\lambda \left(S_\lambda(\boldsymbol{a}^{(K)}) - y_\lambda\right)\delta_{\mu\lambda} z_\nu^{(K-1)} \\ &= \left(S_\mu(\boldsymbol{a}^{(K)}) - y_\mu\right) z_\nu^{(K-1)} \end{aligned} \tag{25}$$

を得る。

なお、式 (25) の左辺を行列 $\partial L/\partial \boldsymbol{W}^{(K)}$ の (μ, ν) 成分とみなすと、右辺が行列

$$\left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)\left(\boldsymbol{z}^{(K-1)}\right)^T = \begin{pmatrix} S_1(\boldsymbol{a}^{(K)}) - y_1 \\ S_2(\boldsymbol{a}^{(K)}) - y_2 \\ \vdots \\ S_r(\boldsymbol{a}^{(K)}) - y_r \end{pmatrix}\left(z_1^{(K-1)}, z_2^{(K-1)}, \cdots, z_m^{(K-1)}\right) \tag{26}$$

の (μ, ν) 成分であることから以下の簡潔な式が得られる ：

$$\frac{\partial L}{\partial \boldsymbol{W}^{(K)}} = \left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)\left(\boldsymbol{z}^{(K-1)}\right)^T \tag{27}$$

ここで、第 $K-1$ 層のユニット数を m とした。また、$\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}$ は（勾配の意味で用いていないので）列ベクトルである。

ここで確率的勾配降下法の考えに基づき、バッチから得られる経験分布 $\hat{q}(\boldsymbol{x}, \boldsymbol{y})$ による L の期待値 $\tilde{J}$ を定義する。

$$\tilde{J}(\boldsymbol{\theta}) = E_{\tilde{q}(\boldsymbol{x}, \boldsymbol{y})}\left[L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})\right] \tag{28}$$

式 (18) と比較すると、$\tilde{J}$ はバッチを用いた損失関数 J の近似値とみなせる。経験分布 $\hat{q}(\boldsymbol{x}, \boldsymbol{y})$ を用いるにあたり、$\boldsymbol{a}^{(K)}$ と $\boldsymbol{z}^{(K-1)}$ については各サンプルについての順伝播で得られた値を用い、$\boldsymbol{y}$ については対応するサンプルの正解値を使う。$\hat{q}(\boldsymbol{x}, \boldsymbol{y})$ は $\boldsymbol{x}$ と $\boldsymbol{y}$ についての分布であるが、現時点でのパラメータ $\boldsymbol{\theta}^{(t)}$ を持つニューラルネットワークと実際の i 番目の入力 $\boldsymbol{x}^{(i)}$ によって決まる $\boldsymbol{a}^{(K,\,i)}$ と $\boldsymbol{z}^{(K-1,\,i)}$、そしてそれに対応する正解値 $\boldsymbol{y}^{(i)}$ を使って期待値を計算するのである。$\boldsymbol{W}^{(K)}$ についてはパラメータベクトルの現在値 $\boldsymbol{\theta}^{(t)}$ から抜き出して使う。ここで

$\tilde{n}$：バッチに含まれるサンプル数
$\boldsymbol{y}^{(i)}$：バッチに含まれる i 番目のサンプルにおける $\boldsymbol{y}$ の正解値
$\boldsymbol{z}^{(K-1,\,i)}$：バッチの i 番目のサンプルの $\boldsymbol{x}^{(i)}$ から順伝播を行った結果得られた $\boldsymbol{z}^{(K-1)}$ の値
$\boldsymbol{a}^{(K,\,i)}$：順伝播の際にサンプル i について計算された活性 $\boldsymbol{a}^{(K,\,i)} = \boldsymbol{W}^{(K)}\boldsymbol{z}^{(K-1,\,i)}$

とすれば、これによって損失関数の微分 $\partial J/\partial W_{\mu\nu}^{(K)}$ の近似値が得られるが、それを $\partial \tilde{J}/\partial W_{\mu\nu}^{(K)}$ で表すことにする。

$$\begin{aligned} \frac{\partial \tilde{J}}{\partial W_{\mu\nu}^{(K)}} &= E_{\tilde{q}(\boldsymbol{x}, \boldsymbol{y})}\left[\left(S_\mu(\boldsymbol{a}^{(K)}) - y_\mu\right) z_\nu^{(K-1)}\right] \\ &= \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\left(S_\mu(\boldsymbol{a}^{(K,\,i)}) - y_\mu^{(i)}\right) z_\nu^{(K-1,\,i)} \end{aligned} \tag{29}$$

ベクトルとしてまとめれば

$$\frac{\partial \tilde{J}}{\partial \boldsymbol{W}^{(K)}} = \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\left(\boldsymbol{S}\left(\boldsymbol{a}^{(K,\,i)}\right) - \boldsymbol{y}^{(i)}\right)\left(\boldsymbol{z}^{(K-1,\,i)}\right)^T \tag{30}$$

と簡潔な式で表せる。

確率的勾配降下法のパラメータ更新に使われるのはこの勾配である。式 (12) により、パラメータの更新は現在のパラメータ値 $\boldsymbol{\theta}^{(t)}$ に対して

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta^{(t)}\left(\nabla_{\boldsymbol{\theta}^{(t)}}\tilde{J}\right)^T \tag{31}$$

として行われる。ここで求めた $\partial \tilde{J}/\partial W_{\mu\nu}^{(K)}$ は $\nabla_{\boldsymbol{\theta}^{(t)}}\tilde{J}$ の成分のひとつである。つまり式 (29) によって重み $W_{\mu\nu}^{(K)}$ がどのように更新されていくかが決まる。

5.3　中間的な重み行列の更新規則

続いて重み行列 $\boldsymbol{W}^{(K-1)}$ についての更新規則を導く。必要なのは $\nabla_{\boldsymbol{\theta}} L = \partial L / \partial \boldsymbol{W}^{(K-1)}$ という勾配である。これが求まれば、バッチから得られる経験分布 $\tilde{q}$ での期待値計算

$$\nabla_{\boldsymbol{\theta}} \tilde{J} = E_{\tilde{q}(\boldsymbol{x},\, \boldsymbol{y})} \left[\nabla_{\boldsymbol{\theta}} L(\boldsymbol{x},\, \boldsymbol{y},\, \boldsymbol{\theta}) \right]$$

により、以下のように確率的勾配降下法で使う勾配が得られる。

$$\frac{\partial \tilde{J}}{\partial W_{\mu\nu}^{(K-1)}} = E_{\tilde{q}(\boldsymbol{x},\, \boldsymbol{y})} \left[\frac{\partial L}{\partial W_{\mu\nu}^{(K-1)}} \right] \tag{32}$$

L は $\boldsymbol{W}^{(K-1)}$ に依存するが、その間には式 (7) と式 (19) で示されるように、複数のベクトル値関数がある。そのためヤコビ行列の連鎖律を繰り返し使って $\partial L / \partial W_{\mu\nu}^{(K-1)}$ を展開する。

$$\frac{\partial L}{\partial W_{\mu\nu}^{(K-1)}} = \frac{\partial L}{\partial \boldsymbol{a}^{(K)}} \frac{\partial \boldsymbol{a}^{(K)}}{\partial \boldsymbol{z}^{(K-1)}} \frac{\partial \boldsymbol{z}^{(K-1)}}{\partial \boldsymbol{a}^{(K-1)}} \frac{\partial \boldsymbol{a}^{(K-1)}}{\partial W_{\mu\nu}^{(K-1)}} \tag{33}$$

式 (33) の最初の因子はスカラー値をもつ L を列ベクトル $\boldsymbol{a}^{(K)}$ で微分しているので L の勾配である。この値は式 (22)、(23) より

$$\frac{\partial L}{\partial a_{\lambda}^{(K)}} = S_{\lambda}(\boldsymbol{a}^{(K)}) - y_{\lambda} \tag{34}$$

または

$$\frac{\partial L}{\partial \boldsymbol{a}^{(K)}} = \Big(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y} \Big)^{T} \tag{35}$$

となる。

第 2 因子 $\partial \boldsymbol{a}^{(K)} / \partial \boldsymbol{z}^{(K-1)}$ は列ベクトル $\boldsymbol{a}^{(K)}$ を列ベクトル $\boldsymbol{z}^{(K-1)}$ で微分しておりヤコビ行列となっている。その $(\lambda,\, \zeta)$ 成分は

$$\begin{aligned} \left(\frac{\partial \boldsymbol{a}^{(K)}}{\partial \boldsymbol{z}^{(K-1)}} \right)_{\lambda\zeta} &= \frac{\partial a_{\lambda}^{(K)}}{\partial z_{\zeta}^{(K-1)}} \\ &= \frac{\partial}{\partial z_{\zeta}^{(K-1)}} \left(\sum_{\gamma} W_{\lambda\gamma}^{(K)} z_{\gamma}^{(K-1)} \right) = W_{\lambda\zeta}^{(K)} \end{aligned} \tag{36}$$

第 3 因子 $\partial \boldsymbol{z}^{(k-1)} / \partial \boldsymbol{a}^{(K-1)}$ については $\boldsymbol{z}^{(K-1)} = \boldsymbol{f}^{(k-1)}(\boldsymbol{a}^{(K-1)})$ に注意して、その $(\zeta,\, \xi)$ 成分を求めれば

$$\begin{aligned} \left(\frac{\partial \boldsymbol{z}^{(K-1)}}{\partial \boldsymbol{a}^{(K-1)}} \right)_{\zeta\xi} &= \left(\frac{\partial \boldsymbol{f}^{(K-1)}(\boldsymbol{a}^{(K-1)})}{\partial \boldsymbol{a}^{(K-1)}} \right)_{\zeta\xi} \\ &= \frac{\partial f_{\zeta}^{(K-1)}(\boldsymbol{a}^{(K-1)})}{a_{\xi}^{(K-1)}} = \left(\Big(\boldsymbol{f}^{(K-1)} \Big)' \right)_{\zeta\xi} \end{aligned} \tag{37}$$

ここで、$(\boldsymbol{f}^{(K-1)})'$ はベクトル値関数 $\boldsymbol{f}^{(K-1)}$ の各成分 $f_{\zeta}^{(K-1)}(\boldsymbol{a}^{(K-1)})$ を $\boldsymbol{a}^{(K-1)}$ の各成分で微分したヤコビ行列である。式 (6) より $\zeta < n$ について $f_{\zeta}^{(K-1)}(\boldsymbol{a}^{(K-1)}) = \psi^{(K-1)}(a_{\zeta}^{(K-1)})$ と定義されており、$f_{\zeta}^{(K-1)}(\boldsymbol{a}^{(K-1)})$ は $a_{\zeta}^{(K-1)}$ 以外に引数を持たない。したがって

$$\Big(\boldsymbol{f}^{(K-1)} \Big)'_{\zeta\xi} = \begin{cases} \Big(\psi^{(K-1)}(a_{\zeta}^{(K-1)}) \Big)' & \zeta = \xi \\ 0 & \zeta \neq \xi \end{cases} \tag{38}$$

ただし ψ' は活性化関数 ψ を微分したものである。

第 4 因子は $\partial \boldsymbol{a}^{(K-1)} / \partial W_{\mu\nu}^{(K-1)}$ は列ベクトルをスカラーで微分しているので列ベクトルであり、式

(24) と同様に

$$\left(\frac{\partial \boldsymbol{a}^{(K-1)}}{\partial W_{\mu\nu}^{(K-1)}}\right)_{\xi} = \frac{\partial a_{\xi}^{(K-1)}}{\partial W_{\mu\nu}^{(K-1)}}$$
$$= \frac{\partial}{\partial W_{\mu\nu}^{(K-1)}}\left(\sum_{\lambda} W_{\xi\lambda}^{(K-1)} z_{\lambda}^{(K-2)}\right) = \delta_{\mu\xi} z_{\nu}^{(K-2)} \tag{39}$$

式 (34) の行ベクトル、(36)、(37) の行列、(39) の列ベクトルを式 (33) へ代入すれば

$$\frac{\partial L}{\partial W_{\mu\nu}^{(K-1)}} = \frac{\partial L}{\partial \boldsymbol{a}^{(K)}}\frac{\partial \boldsymbol{a}^{(K)}}{\partial \boldsymbol{z}^{(K-1)}}\frac{\partial \boldsymbol{z}^{(K-1)}}{\partial \boldsymbol{a}^{(K-1)}}\frac{\partial \boldsymbol{a}^{(K-1)}}{\partial W_{\mu\nu}^{(K-1)}}$$
$$= \sum_{\lambda,\zeta,\xi}\left(S_{\lambda}(\boldsymbol{a}^{(K)}) - y_{\lambda}\right) W_{\lambda\zeta}^{(K)}\left(\left(\boldsymbol{f}^{(K-1)}\right)'\right)_{\zeta\xi}\delta_{\mu\xi} z_{\nu}^{(K-2)} \tag{40}$$

$((\boldsymbol{f}^{(K-1)})')_{\zeta\xi}\delta_{\mu\xi}$ により、総和 $\sum_{\zeta}\sum_{\xi}$ のうち、$\zeta = \xi = \mu$ となる項しか残らないので

$$\frac{\partial L}{\partial W_{\mu\nu}^{(K-1)}} = \sum_{\lambda}\left(S_{\lambda}(\boldsymbol{a}^{(K)}) - y_{\lambda}\right) W_{\lambda\mu}^{(K)}\left(\psi^{(K-1)}(a_{\mu}^{(K-1)})\right)' z_{\nu}^{(K-2)}$$
$$= \left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)^{T} \boldsymbol{W}_{:\mu}^{(K)}\left(\psi^{(K-1)}(a_{\mu}^{(K-1)})\right)' z_{\nu}^{(K-2)} \tag{41}$$

と表すことができる。ただし $\boldsymbol{W}_{:\mu}^{(K)}$ は重み行列 $\boldsymbol{W}^{(K)}$ の第 μ 列である。

ここで

$$\left(\left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)^{T} \boldsymbol{W}^{(K)}\left(\boldsymbol{f}^{(K-1)}\right)'\right)_{\mu} = \sum_{\alpha}\left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)_{\alpha}\left(\boldsymbol{W}^{(K)}\left(\boldsymbol{f}^{(K-1)}\right)'\right)_{\alpha\mu}$$
$$= \sum_{\alpha,\,\beta}\left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)_{\alpha} W_{\alpha\beta}^{(K)}\left(\boldsymbol{f}^{(K-1)}\right)'_{\beta\mu}$$
$$= \sum_{\alpha,\,\beta}\left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)_{\alpha} W_{\alpha\beta}^{(K)}\delta_{\beta\mu}\left(\psi^{(K-1)}(a_{\mu}^{(K-1)})\right)'$$
$$= \sum_{\alpha}\left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)_{\alpha} W_{\alpha\mu}^{(K)}\left(\psi^{(K-1)}(a_{\mu}^{(K-1)})\right)'$$
$$= \left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)^{T} \boldsymbol{W}_{:\mu}^{(K)}\left(\psi^{(K-1)}(a_{\mu}^{(K-1)})\right)' \tag{42}$$

となることに注意すれば、式 (41) は

$$\frac{\partial L}{\partial W_{\mu\nu}^{(K-1)}} = \left(\left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)^{T} \boldsymbol{W}^{(K)}\left(\boldsymbol{f}^{(K-1)}\right)'\right)_{\mu} z_{\nu}^{(K-2)} \tag{43}$$

と表すことができる。

次に真の分布 $q(\boldsymbol{x}, \boldsymbol{y})$ による期待値の代わりに、バッチから得られる経験分布 $\tilde{q}$ を使って期待値を求める。

$$\frac{\partial \tilde{J}}{\partial W_{\mu\nu}^{(K-1)}} = E_{\tilde{q}(\boldsymbol{x},\,\boldsymbol{y})}\left[\frac{\partial L}{\partial W_{\mu\nu}^{(K-1)}}\right]$$
$$= E_{\tilde{q}(\boldsymbol{x},\,\boldsymbol{y})}\left[\left(\left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y}\right)^{T} \boldsymbol{W}^{(K)}\left(\boldsymbol{f}^{(K-1)}\right)'\right)_{\mu} z_{\nu}^{(K-2)}\right]$$
$$= \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\left(\left(\boldsymbol{S}(\boldsymbol{a}^{(K,\,i)}) - \boldsymbol{y}^{(i)}\right)^{T} \boldsymbol{W}^{(K)}\left(\boldsymbol{f}^{(K-1)}\right)'\right)_{\mu} z_{\nu}^{(K-2,\,i)} \tag{44}$$

5.4 逆伝播で誤差を伝える

第 $K-2$ 層についても連鎖律により勾配を求めることができる。

$$\frac{\partial L}{\partial W_{\mu\nu}^{(K-2)}} = \frac{\partial L}{\partial \boldsymbol{a}^{(K)}} \frac{\partial \boldsymbol{a}^{(K)}}{\partial \boldsymbol{z}^{(K-1)}} \frac{\partial \boldsymbol{z}^{(K-1)}}{\partial \boldsymbol{a}^{(K-1)}} \frac{\partial \boldsymbol{a}^{(K-1)}}{\partial \boldsymbol{z}^{(K-2)}} \frac{\partial \boldsymbol{z}^{(K-2)}}{\partial \boldsymbol{a}^{(K-2)}} \frac{\partial \boldsymbol{a}^{(K-2)}}{\partial W_{\mu\nu}^{(K-2)}} \tag{45}$$

$$= \frac{\partial L}{\partial \boldsymbol{a}^{(K-1)}} \frac{\partial \boldsymbol{a}^{(K-1)}}{\partial \boldsymbol{z}^{(K-2)}} \frac{\partial \boldsymbol{z}^{(K-2)}}{\partial \boldsymbol{a}^{(K-2)}} \frac{\partial \boldsymbol{a}^{(K-2)}}{\partial W_{\mu\nu}^{(K-2)}} \tag{46}$$

このうち、式 (45) の右辺の先頭の 3 つの因子は $\partial L/\partial W_{\mu\nu}^{(K-1)}$ を求めるときにすでに計算している。つまり 5.3 項での計算結果を再利用できる。誤差逆伝播法では出力に近い側から勾配の計算をスタートしているの、それより遠い層についての勾配計算に使えるのである。式 (45) の右辺の 3 つの因子の積は連鎖律により $\partial L/\partial \boldsymbol{a}^{(K-1)}$ とも書ける。5.3 項の計算を行った際、これを残しておくとよい。最後の 3 つの因子に関しても、5.3 項の方法で、（ただし添字を変えて）求められる。

第 $K-3$ 層から第 1 層についても、同様の形で $\partial L/\partial W_{\mu\nu}^{(l)}$ の計算が行える。連鎖律を使うと、そのほとんどの因子は出力に近い側の層での微分の際にすでに求められているので、それに層 l についての微分を掛けるだけである。目的とする微分である $\partial L/\partial W_{\mu\nu}^{(l)}$ に対して連鎖律を使って展開し、式 (36)、(37)、(39) の K や $K-1$ を l や $l-1$ に書き換えて使うことができる。

勾配 $\partial L/\boldsymbol{a}^{(l)}$ は行ベクトル、$\partial \boldsymbol{a}^{(l)}/\partial W_{\mu\nu}^{(l)}$ は列ベクトルであることに注意して、式 (39) を用いれば

$$\frac{\partial L}{\partial W_{\mu\nu}^{(l)}} = \frac{\partial L}{\partial \boldsymbol{a}^{(l)}} \frac{\partial \boldsymbol{a}^{(l)}}{\partial W_{\mu\nu}^{(l)}} = \sum_{\gamma} \left(\frac{\partial L}{\partial \boldsymbol{a}^{(l)}} \right)_{\gamma} \delta_{\mu\gamma} z_{\nu}^{(l-1)} = \left(\frac{\partial L}{\partial \boldsymbol{a}^{(l)}} \right)_{\mu} z_{\nu}^{(l-1)} \tag{47}$$

を得る。また、式 (36)、(37) に注意して、連鎖律を使えば

$$\frac{\partial L}{\partial \boldsymbol{a}^{(l-1)}} = \frac{\partial L}{\partial \boldsymbol{a}^{(l)}} \frac{\partial \boldsymbol{a}^{(l)}}{\partial \boldsymbol{z}^{(l-1)}} \frac{\partial \boldsymbol{z}^{(l-1)}}{\partial \boldsymbol{a}^{(l-1)}} = \frac{\partial L}{\partial \boldsymbol{a}^{(l)}} \boldsymbol{W}^{(l)} \left(\boldsymbol{f}^{(l-1)} \right)' \tag{48}$$

となる。式 (48) では $\partial L/\boldsymbol{a}^{(l)}$ を使って $\partial L/\boldsymbol{a}^{(l-1)}$ を再帰的に定義している。式 (23) の

$$\frac{\partial L}{\partial \boldsymbol{a}^{(K)}} = \left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y} \right)^T \tag{49}$$

から出発し、l を減らしながら繰り返し式 (48) を使っていけば、$l=1$ までのすべての $\partial L/\boldsymbol{a}^{(l)}$ が求まる。これがまさしく逆伝播である。

5.5 デルタの更新規則

式 (49) は予測値ベクトル $\boldsymbol{S}(\boldsymbol{a}^{(K)})$ と正解値ベクトル $\boldsymbol{y}$ の差であり、誤差ベクトルと言える。この誤差ベクトルを $\boldsymbol{\delta}$ と表す。

$$\boldsymbol{\delta}^{(K)} = \frac{\partial L}{\partial \boldsymbol{a}^{(K)}} = \left(\boldsymbol{S}(\boldsymbol{a}^{(K)}) - \boldsymbol{y} \right)^T \tag{50}$$

誤差逆伝播法は誤差ベクトル $\boldsymbol{\delta}^{(K)}$ を多数のヤコビ行列によって変換して入力側に伝えているとも解釈できる。そこで、ヤコビ行列による変換後の誤差ベクトル、すなわち一般の l についてのデルタ $\boldsymbol{\delta}^{(l)}$ を以下のように定義する。

$$\boldsymbol{\delta}^{(l)} = \frac{\partial L}{\partial \boldsymbol{a}^{(l)}} \tag{51}$$

すなわちデルタは活性による L の勾配である。式 (51) の右辺はまさに式 (48) で再帰的に定義された勾配 $\partial L/\boldsymbol{a}^{(l)}$ であるので、デルタについて以下の再帰的な定義が得られる。

$$\boldsymbol{\delta}^{(l-1)} = \boldsymbol{\delta}^{(l)} \boldsymbol{W}^{(l)} \left(\boldsymbol{f}^{(l-1)} \right)' \tag{52}$$

これは $\boldsymbol{\delta}^{(l)}$ に右から $\boldsymbol{W}^{(l)} \left(\boldsymbol{f}^{(l-1)} \right)'$ を掛けるという変換を行うことで $\boldsymbol{\delta}^{(l-1)}$ が得られることを表わしている。l は層の番号を表すので、$\boldsymbol{\delta}^{(l)}$ から $\boldsymbol{\delta}^{(l-1)}$ に進むことは入力側に近づくことを意味する。こ

の再帰的な定義をデルタの更新規則（delta rule）と呼ぶ。誤差逆伝播法とうい名称も誤差を変換して伝えていると捉えられることからきている。

デルタが求まれば式 (47) と式 (51) からパラメータ更新のための勾配がもとまる。

$$\frac{\partial L}{\partial W_{\mu\nu}^{(l)}} = \boldsymbol{\delta}_{\mu}^{(l)} z_{\nu}^{(l-1)} \tag{53}$$

$\boldsymbol{\delta}^{(l)}$ は勾配であるので、行ベクトルであることから、式 (53) の右辺は行列

$$\left(\boldsymbol{\delta}^{(l)}\right)^{T}\left(\boldsymbol{z}^{(l-1)}\right)^{T} = \begin{pmatrix} \delta_1^{(l)} \\ \delta_2^{(l)} \\ \vdots \end{pmatrix} \begin{pmatrix} z_1^{(l-1)} & z_2^{\prime l-1)} & \cdots \end{pmatrix}$$

の (μ, ν) 成分とみなせる。式 (53) の左辺を行列 $\partial L/\partial \boldsymbol{W}^{(l)}$ の (μ, ν) 成分とみなすと、以下の簡潔な表記も得られる。

$$\frac{\partial L}{\partial \boldsymbol{W}^{(l)}} = (\boldsymbol{\delta}^{(l)})^{T}(\boldsymbol{z}^{(l-1)})^{T} \tag{54}$$

各 l についての $\partial L/\partial \boldsymbol{W}^{(l)}$ を求めるにあたり、逆伝播では $\boldsymbol{\delta}^{(l)}$ を保存して再利用するだけでよく、ヤコビ行列の保存は必要ないので、記憶容量の点で大幅にメリットがある。

確率的勾配降下法を使った場合の誤差逆伝播法のアルゴリズムの擬似コードを以下に示す。ただし、サンプル i についての活性を $\boldsymbol{a}^{(l,i)}$ 勾配を $(\partial L/\partial \boldsymbol{a}^{(K)})^{(i)}$ のように表している。

アルゴリズム 1 確率的勾配降下法による誤差逆伝播法

入力：入力 $\boldsymbol{x}^{(i)}$ と正解値 $\boldsymbol{y}^{(i)}$ のペアの集合

出力：各 l についての重み行列 $\boldsymbol{W}^{(l)}$

各 l についての重み行列 $\boldsymbol{W}^{(l)}$ を初期化

while 損失関数 $\tilde{J}$ が目標値以上

訓練データの中からバッチを取り出す。

バッチに属す $\left\{\left(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}\right), \cdots, \left(\boldsymbol{x}^{(\tilde{n})}, \boldsymbol{y}^{(\tilde{n})}\right)\right\}$ について以下を繰り返す。

順伝播：

for $i = 1$ **to** $\tilde{n}$

$\boldsymbol{z}^{(0,i)} = \boldsymbol{x}^{(i)}$

for $l = 1$ **to** K

$\boldsymbol{a}^{(l,i)} = \boldsymbol{W}^{(l)}\boldsymbol{z}^{(l-1,i)}$ を計算する。

$\boldsymbol{z}^{(l,i)} = \boldsymbol{f}^{(l)}(\boldsymbol{a}^{(l,i)})$ を計算する。　[各層の出力の計算]

逆伝播：

for $i = 1$ **to** $\tilde{n}$

$\boldsymbol{\delta}^{(K,i)} = \left(\partial L/\partial \boldsymbol{a}^{(K)}\right)^{(i)}$ を計算する。　[デルタの計算]

for $l = K$ **to** 1

$\left(\partial L/\partial \boldsymbol{W}^{(l)}\right)^{(i)} = \left(\boldsymbol{\delta}^{(l,i)}\right)^{T}\left(\boldsymbol{z}^{(l-1,i)}\right)^{T}$ を計算する。　[勾配の計算]

$\boldsymbol{\delta}^{(l-1,i)} = \boldsymbol{\delta}^{(l,i)}\boldsymbol{W}^{(l)}\left(\boldsymbol{f}^{(l)}\right)'$ を計算する。　[デルタの更新]

$\partial \tilde{J}/\partial \boldsymbol{W}^{(l)} = (1/\tilde{n})\sum_{i=1}^{\tilde{n}}\left(\partial L/\partial \boldsymbol{W}^{(l)}\right)^{(i)}$ を計算する。

重み行列 $\boldsymbol{W}^{(l)}$ から $\eta\partial \tilde{J}/\partial \boldsymbol{W}^{(l)}$ を引く。　[パラメータ更新]

なお、損失関数としてクロスエントロピーを使わない場合、あるいは出力関数としてソフトマックス関数を使わない場合も、誤差逆伝播法は同様に使用できる。$\partial L/\partial \boldsymbol{a}^{(K)}$ が変わるだけであり、残りの計算は同じである。

5.6　勾配消失問題

デルタの更新規則を表す (52) では右辺で $(\boldsymbol{f}^{(l)})'$ が掛けられているが、$(\boldsymbol{f}^{(l)})'$ の成分は $(\psi^{(l)})'$ と 0 である。もし活性化関数としてシグモイド関数

$$\psi^{(l)}(a) = \sigma(a) = \frac{1}{1+\exp(-a)}$$

を使用した場合、シグモイド関数の微分は

$$\sigma'(a) = \sigma(a)(1-\sigma(a))$$

であるため、$a \to \infty$ で $\sigma(a) = 1$ でも、$a \to -\infty$ で $\sigma(a) = 0$ でも微分 $\sigma'(a)$ は 0 に近づく。このように引数の値が大きくあるいは小さくなるにつれて、微分が 0 に近づく性質を飽和的非線形性（saturating nonlinearlity）と呼ぶ。

デルタは再帰的に定義され、繰り返し ψ' が掛けられるため、出力層から離れるに従い $\boldsymbol{\delta}^{(l)}$ が急速に小さくなっていく。パラメータ更新のための勾配 $\partial\tilde{J}/\partial\boldsymbol{W}$ はデルタから求まるので、出力層から離れた隠れ層の重み行列はほとんど更新されなくなる。これは勾配消失問題（vanishing gradient problem）と呼ばれ、長年ニューラルネットワークの研究者を悩ませてきた課題であった。層の数が増えるにつれてデルタが減衰していくため、浅いネットワークネットワークでは問題にならないが、深いネットワークに学習させることが困難であった。しかしこの問題は活性化関数として ReLU (rectified linear unit)

$$\mathcal{R}(a) = \begin{cases} a & (a \geq 0) \\ 0 & (a < 0) \end{cases} \tag{55}$$

を使うことであっさり解決した。ReLU の微分は

$$\frac{d\mathcal{R}(a)}{da} = \begin{cases} 1 & (a < 0) \\ \text{undefined} & (a = 0) \\ 0 & (a < 0) \end{cases} \tag{56}$$

であり、正の領域では微分は 1 であり、引数がいくら大きくなっても 0 に近づかないからである。これは飽和的非線形性を持たないということである。活性化関数に ReLU を使用して勾配消失問題を解決したことは、深層学習の爆発的な発展の一因である。

なお、再帰型ニューラルネットワークの場合、活性化関数の微分ではなく重み行列を繰り返し掛けることで生じる勾配消失問題が存在し、別の形で対策が行われている。

5.7　順伝播と逆伝播の反復

逆伝播のフェーズでは $\partial\tilde{J}/\partial\boldsymbol{W}^{(K)}$ から $\partial\tilde{J}/\partial\boldsymbol{W}^{(1)}$ までを求め、重み行列の更新を行う。それが終わった後、別のバッチ（サンプル集合）を使い、ふたたび順伝播を行う。それが出力ユニットに達したら、ふたたび逆伝播が行われる。これを繰り返して重み行列を更新していくことが誤差逆伝播法である。

各バッチを構成するサンプルは訓練データの集合から順番に取り出されることも多い。それでも確率的勾配降下法が確率的なアルゴリズムとみなせるのは、訓練データをあらかじめシャッフル（ランダムに並び替え）しておくことが多いためである。

バッチを次々に取り出していき、訓練データのすべてを使い終えたとき、ひとつのエポック（epoch）が終わったと表現する。確率的勾配降下法ではあらかじめ定めた条件を満たすまでエポックを繰り返してパラメータを更新する。深層学習ではどれだけ学習を行ったかの度合いをエポック数で表現することが多い。

ここでは、多層ニューラルネットワークを例にして誤差逆伝播法を説明したが、一般の順伝播型ネットワークにおいても同じように出力に近い側から微分を計算し、遠いものに向かってデルタを伝えていくという流れは同じである。