バックプロパゲーション

浅川伸一 <asakawa@twcu.ac.jp>

1　バックプロパゲーション (誤差逆伝播法)

1.1　XOR 問題、線形分離不可能な問題

パーセプトロンでは絶対に解けない問題に排他的論理和 (XOR) 問題がある。排他的論理和とは、2 つの入力のうちいずれか一方のみが 1 のとき 1 を出力する問題である。図 1 左を見るとわかるとおり、XOR 問題は一本の判別直線で白マルと黒マルを分離できない、すなわち線形分離不可能な問題である。

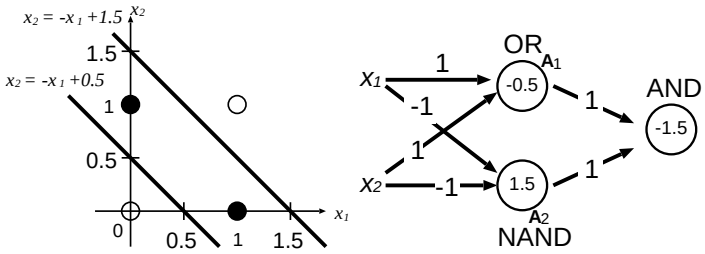

図 1: XOR 問題の幾何学的表現と XOR 問題を解くためのネットワーク

図 1 右は図 1 左の幾何学表現を対応するネットワークとして表現したものである。一番左が入力層、一番右が出力層、中間の 2 つが中間層である。ユニットの中に書かれた数値は各ユニットのしきい値を示している。中間層の 2 つのユニットのうち上の OR と書かれたユニットは、$x_1 + x_2 - 0.5 > 0$ のとき発火する。この式を書き換えると、$x_2 > -x_1 + 0.5$ となるので図 1 左の下斜線より上の領域に対応する。一方、中間層の下の NAND(not and) と書かれたユニットは、$-x_1 - x_2 + 1.5 > 0$ のとき発火するので、移項して $x_2 < -x_1 - 1.5$ とすれば、図 1 左の上斜線より下の領域に対応していることが分かる。さらに、AND と書かれた出力ユニットは、2 つの中間層ユニットからの情報の論理積 (AND) を計算するユニットになっている。そこで、2 つの中間層ユニットの両方が発火する場合だけ、出力ユニットも発火し、1 を出力する。これは、図 1 左では、「下の斜線より上の領域」でかつ「上の斜線よ

り下の領域」に対応する。すなわち、図中の黒丸の領域だけが分離されることになる。このような 2 本の直線は図中にいくらでも引けることから、XOR 問題の解も無数に存在することが分かる。

入力層		中間層		出力
x_1	x_2	a_1	a_2	r
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

表 1: 図 1 に対応する XOR 問題の真偽表

図 1 左にあるとおり、中間層のユニット 1 個は 1 つの線形判別関数に相当すると考えられる。中間層から出力層への結合では各出力の論理積 AND を計算していることに相当する。n 個の中間層を用意すれば原理的には $\frac{n^2+n+2}{2}$ 個のカテゴリー分類が可能である。パーセプトロンが XOR 問題を解くことができない理由は入力層から中間層にいたる結合係数を変更する手段がないことなのである。

1.2 誤差逆伝播法 (一般化デルタルール)

XOR 問題でも見たように、パーセプトロンの問題点は学習が出力層と中間層の間だけで行われ、入力層と中間層の結合係数を更新することができない。換言すれば中間層ユニットの誤差を明示的に知る方法はない。このことは信用割り当て credit assignment 問題と呼ばれる。この信用割り当て問題を解決したのがバックプロパゲーション (誤差逆伝播法あるいは一般化デルタルール) である。

m 層のネットワークを考え、k 層の i 番目のユニットへの総入力を x_i^k, このユニットの出力を y_i^k, $k-1$ 層の i 番目のユニットから k 層の j 番目のユニットへの結合係数を $w_{ij}^{k-1,k}$ と表記する。各ユニットの出力は

$$y_i^k = f\left(x_i^k\right) = \frac{1}{1+e^{-x_i^k}} \tag{1}$$

$$x_j^k = \sum_i w_{ij}^{k-1,k} y_i^{k-1}, \tag{2}$$

で定義されているものとする。あるデータ $\boldsymbol{x}$ と教師信号 $\boldsymbol{t}$ が与えられたとき教師信号と出力との 2 乗誤差を

$$E = \frac{1}{2}\sum_j \left(y_j^m - t_j\right)^2 = \frac{1}{2}\sum_j \left(\delta_j^m\right)^2 \tag{3}$$

と表記する。加算記号 $\sum$ の前の 1/2 は微分したときに式を簡単にする程度の意味しかないので本質的ではない。この誤差関数 E は、教師信号と出力との差の 2 乗に比例して大きくなる。そこで、E が減少する方向に $\boldsymbol{w}$ の値を逐次更新することがバックプロパゲーション法である。(3) 式の誤差 E は各 y_j^m の 2 次関数とみなすことができるので $E \geq 0$ であり、$E = 0$ となるのは、すべての y_j^m に対して $y_j^m - t_j = 0$ のとき、すなわち完全に学習が成立したときのみである。

結合係数の更新式を

$$\Delta w_{ij}^{k-1,k} = -\epsilon \frac{\partial E}{\partial w_{ij}^{k-1,k}} \tag{4}$$

とするのがバックプロパゲーション法における学習である。(4) 式の右辺で誤差 E を $w_{ij}^{k-1,k}$ で微分しなければならない。これは

$$\frac{\partial E}{\partial w_{ij}^{k-1,k}} = \frac{\partial E}{\partial y_j^m} \frac{\partial y_j^m}{\partial w_{ij}^{m-1,m}} = \sum_j \delta_j^m \frac{\partial y_j^m}{\partial w_{ij}^{m-1,m}} \tag{5}$$

のように書くことができる。これを合成関数の微分公式あるいはチェーンルールと言ったりする。$\partial y/\partial w$ は、もう一度合成関数の微分の公式を使って

$$\frac{\partial y_j^m}{\partial w_{ij}^{m-1,m}} = \frac{\partial y_j^m}{\partial x_j^m} \frac{\partial x_j^m}{\partial w_{ij}^{m-1,m}} \tag{6}$$

と書くことができる。換言すれば (4) 式で E は y_j^m の関数であるが、さらに y_j^m は x_j^m の関数であり、さらにさらに x_j^m は $w_{ij}^{m-1,m}$ の関数である。$\partial y/\partial x$ はシグモイド関数なので

$$\frac{\partial y}{\partial x} = y\,(1-y) \tag{7}$$

であり、$\partial x_j^m / \partial w_{ij}^{m-1,m} = y_i^{m-1}$ である。これらをまとめて書くと

$$\begin{aligned}
\Delta w_{ij}^{m-1,m} &= -\epsilon \frac{\partial E}{\partial w_{ij}^{m-1,m}} && (8)\\
&= -\epsilon \frac{\partial E}{\partial y_j^m} \frac{\partial y_j^m}{\partial x_j^m} \frac{\partial x_j^m}{\partial w_{ij}^{m-1,m}} && (9)\\
&= -\epsilon \sum_j \left(y_j^m - t_j\right) y_j^m \left(1 - y_j^m\right) y_i^{m-1} && (10)\\
&= -\epsilon \sum_j \delta_j^m y_j^m \left(1 - y_j^m\right) y_i^{m-1}. && (11)
\end{aligned}$$

となる。ここで、$\delta_j^m = \left(y_j^m - t_j\right)$ である。もし仮に (1) 式で与えられている出力関数が線形関数 $y(x) = x$ であれば、(11) 式は $\Delta w_{ij} = (t_j - y_j)\, y_i^{m-1}$ となってパーセプトロンの学習則と一致する。以上が最上位層 m とそのすぐ下の $m-1$ 層との結合係数の更新式である。

次に中間層以下第 n 層 $(n \neq m)$ のユニット y_j^n の結合係数の更新を考えるために、誤差 E をそのユニット y_j^n で微分する。

$$\frac{\partial E}{\partial y_j^n} = \sum_k \frac{\partial E}{\partial y_k^{n+1}} \frac{\partial y_k^{n+1}}{\partial x_k^{n+1}} \frac{\partial x_k^{n+1}}{\partial y_j^n} \tag{12}$$

$$= \sum_k \frac{\partial E}{\partial y_k^{n+1}} \frac{\partial y_k^{n+1}}{\partial x_k^{n+1}} \frac{\partial}{\partial y_j^n} \sum_i w_{ki}^{n,n+1} y_i^n \tag{13}$$

$$= \sum_k \delta_k^{n+1} y_k^{n+1} \left(1 - y_k^{n+1}\right) w_{kj}^{n,n+1}, \tag{14}$$

この式から中間層ユニット y_j^n が、出力層での誤差 E にどのような影響を与えるのかが分かる。すなわち、中間層のユニット y_j^n の影響は、上の層のユニットの誤差 δ_k^{n+1} に、その中間層ユニットから上の層のユニット y_k^{n+1} への結合係数 $w_{kj}^{n,n+1}$ をかけ、さらに $\partial y/\partial x = y(1-y)$ をかけたものになることが分かる。この量 δ_j^n を信用割り当て credit assignment として学習すれば良いことになる。この計算は再帰的に逐次下の層のユニットへ誤差信号 δ を計算することができる。以上をまとめると、結合係数の修正量 $w_{ij}^{k-1,k}$ は

$$\Delta w_{ij}^{k-1,k} = -\epsilon \delta_j^k y_j^k \left(1 - y_j^k\right) y_i^{k-1}. \tag{15}$$

となる。ここで、

$$\delta_j^k = \begin{cases} \left(y_j^k - t_j\right), & \text{if } k = m \\ \left(\sum_i \delta_l^{k+1} w_{ji}^{k,k+1}\right), & \text{otherwise} \end{cases} \tag{16}$$

である。式 (16) を見ると誤差の計算がネットワークの出力を求める計算と逆の流れで入力層まで伝播するようになっていることがわかる。これが誤差逆伝播法と呼ばれる所以である。

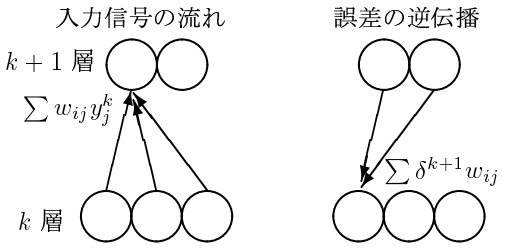

図 2: 誤差逆伝播の模式図

式 (16) から分かることがもう一つある。それは学習を始める前に w を乱数で初期化しなければならないことである。実際に $w = 0$ であれば誤差がゼロになってしまい学習が進まない。ゆえに w の初期値は乱数を用いて何度かくり返して学習させることが必要である。

1.3 実習：排他的論理和の学習パターン

何度か XOR 問題を解かせてみると大まかに 3 パターンの学習曲線を描くことができる。

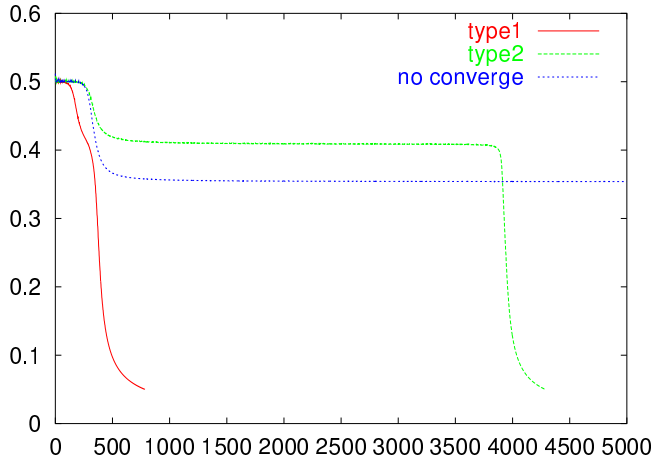

図 3: 典型的な学習パターン

上図は横軸に学習回数、縦軸に平均二乗誤差をプロットしたものである。異なる初期値で program を動かすと、速やかに学習が終了するもの、error が 0.4 付近まで減少したあと学習が進まないもの、error が 0.4 付近で一旦止まって、その後急激に学習が進むもの、などが観察できる。

XOR 問題が解けるか解けないか、どのパターンを示すかは初期値によって決まる。そこで、どのような初期値の時にどのパターンになるかを調べてみよ。

1.4 実習：関数近似

任意の関数が パーセプトロンで表現できることを説明した。そこでこのことを確かめてみよう。つぎのような関数

$$f(x) = \sin(4x\pi)\exp(-2\pi x^2)/\pi + 0.5 \tag{17}$$

を考えることにする。この関数は Gabor 関数と呼ばれる関数の変形である。最後に $+0.5$ してあるのは、$f(x)$ の値が $[0, 1]$ の間に納まるようにするためである。Gabor 関数は第一次視覚野の受容野特性を記述したりするのに用いられる心理学では有名な関数である。Gabor 関数は以下のようなグラフになる。図中の $+$ で表した点でこの関数を近似するとすると、点は

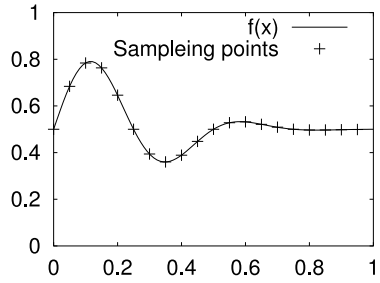

図 4: Gabor 関数

```
x 座標   f(x)
0.000    0.500
0.050    0.684
0.100    0.784
0.150    0.763
0.200    0.646
0.250    0.500
0.300    0.394
0.350    0.360
0.400    0.389
0.450    0.448
0.500    0.500
0.550    0.528
0.600    0.532
0.650    0.521
0.700    0.509
0.750    0.500
0.800    0.497
0.850    0.497
0.900    0.498
0.950    0.499
1.000    0.500
```

となる。前述のとおり一個の中間層で一つの判別関数が実現できるのだから 10 個の点を用いると 10 個の中間層を用意すれば学習できるはずである。ここでは、中間層を 4 個にしてみて BP に学習させた。ちなみに結果 (結合係数) は

```
###  hidden[1]
```

```
 +0.824219  -4.218125
###  hidden[2]
 -0.295888  -0.843664
###  hidden[3]
 -0.275941  -0.953814
###  hidden[4]
 -0.107143  -1.499846
###  output[1]
 +0.037168  +0.795574  -0.650240  -0.594141  -1.006148
```

のようになった。ここで、中間層のニューロンの役割を確認するために、同じグラフに中間層のニューロンの出力を同じグラフに重ね書きしてみよう。入力層のニューロンが 1 個しかないので、中間層の出力も

$$f(x) = 1/(1 + \exp(-(\text{入力層からの重み係数} + \text{しきい値}) \times x)) \qquad (18)$$

になるので、同じグラフに重ね書きできる。4 つの中間層で実行してみた結果を同じグラフ上に plot すると、のようになる。このグラフを見ると、二

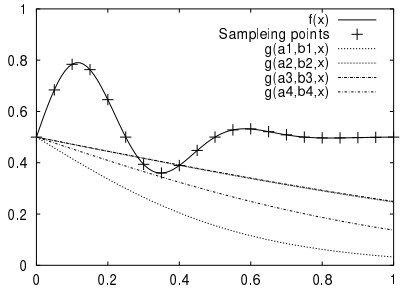

図 5: 中間層の出力と重ね書き

番目の中間層ニューロンと三番目の中間層ニューロンの出力グラフがほぼ重なっている。ということは、この関数は中間層ニューロンの数が 3 個でも良かったことを表してる。ちなみにこのグラフは gnuplot で作成した。サンプルオペレーションを以下に載せる。

```
gnuplot> set xrange [0 : 1]
gnuplot> set yrange [0 : 1]
gnuplot> f(x) = sin(4 * x * pi ) * exp( -2 * pi * x ) + 0.4
gnuplot> g(alpha,beta,x)=1/(1+exp(- (alpha + beta ) * x))
gnuplot> a1 = -6.09
gnuplot> a2 = -3.76
gnuplot> a3 = -3.54
```

```
gnuplot> a4 = -1.01
gnuplot> b1 = 1.0
gnuplot> b2 = -0.12
gnuplot> b3 = -0.15
gnuplot> b4 = -0.34
gnuplot> plot f(x), "gabor.data" title "Samplein points", \
        g(a1,b1,x), g(a2,b2,x), g(a3,b3,x), g(a4,b4,x)
```

$[0,1]$ の範囲で定義され、かつ $f(x)$ の値も $[0,1]$ に納まるような適当な関数 (例えば $y = x^2$) を一つ考えて、その関数の値を使って入力層 1, 中間層 n, 出力層 1 の BP のプログラムに学習させてみよ。

また、中間層の数を変化させたときに学習回数がどうなるかを調べてみよ。

1.5 実習：もう一つの XOR 問題の解

排他的論理和 XOR 問題には最低でも 2 つの中間層ユニットが必要であることを説明した。だがこれは完全な階層型のネットワークの場合のみであり、入力層から出力層への直接結合がある場合には、中間層ユニットは 1 つでもよい。

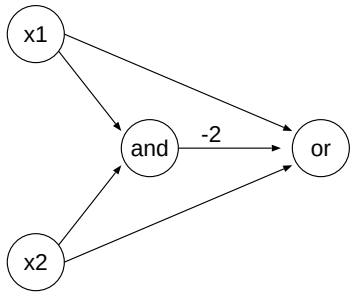

図 6: 中間層のユニット数が 1 で排他的論理和問題を解くネットワーク

図から分かるとおり、一つだけしかない中間層ユニットは二つの入力の論理積 AND を計算しており、その中間層ユニットからの出力が負で出力層ユニットに結合している。すなわち入力が論理積のとき中間層ユニットから出力層ユニットへ強い抑制がかかることを意味している。出力層ユニットは入力層からの直接結合において論理和 OR を計算している。従って入力層ユニット x_1, x_2 のいずれかからの信号を受けて活性化する。ただし x_1, x_2 の両方の入力層ユニットが活性化した場合だけは 1 つの中間層ユニットからの強い抑制性の信号により活動が抑えられる。

さまざまな課題を与えて完全な階層型のアーキテクチュアと入力層から出力層への直接結合があるアーキテクチュアの比較を行なえ。

2 バックプロパゲーション法の応用例

2.1 対称性の認識

BP 法の成功例として次に示す対称性を認識するネットワークを挙げる。図

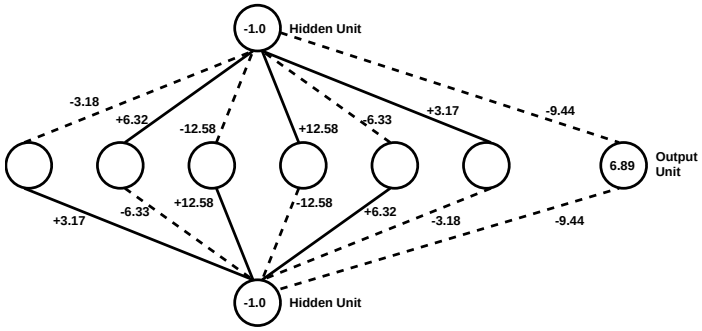

図 7: 対称性を認識するネットワーク (PDP book p.340)

中の点線は負の結合を、実線は正の結合係数を表す。また、円の中に書かれた数字はしきい値である。図を見ると分かるとおり中間層への結合係数が綺麗に対称になっており二つの中間層ユニットで正負が逆転していることが分かる。

2.2 実習：対称性の認識

対象性を学習するバックプロパゲーションのプログラムで、学習セットをいくつか取り除いて学習させよ。学習成立後、取り除いたパターンを用いて一般化能力を試せ。

全 64 個の学習パターン中、対称なパターンを 1 個、非対称なパターンを 8 個取り除き学習させよ。学習後に取り除いた 9 個のパターンを入力すると正解が得られるかどうかを調べよ。

同様に、対称なパターンを 2 個、非対称なパターンを 16 個取り除き学習させよ。学習後に取り除いた 18 個のパターンを入力すると正解が得られるかどうかを調べよ。

対称性を学習するネットワークでは、最低何個の学習パターンが必要かを検討せよ。

2.3 砂時計モデル

バックプロパゲーションの学習アルゴリズムにの応用例として、恒等写像の学習による情報圧縮がある。恒等写像の学習とは、入力データと教師信号を同じくして学習させることである。中間層のユニット数が入力層ユニットや出力層ユニットに比べて少ないとき、中間層ユニットにおいてデータ圧縮表現が得られる。

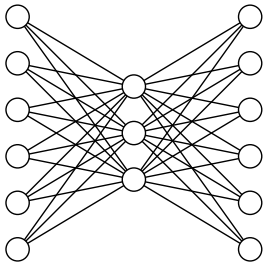

図 8: 砂時計モデル

2.4 実習：データ圧縮

128×128 ピクセルの一枚の画像を 8×8 の小領域に分割し、恒等写像を学習させた。恒等写像なので入力ベクトルも出力ベクトルも同じ 64 次元である。ここでは中間層のユニット数を 16 とした。すなわち $16/64 = 1/4$ のデータ圧縮を行なったことになる。

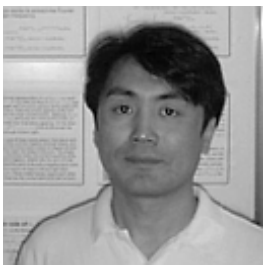

図 9: 恒等写像を行なった原画像

一ピクセルあたり 256 階調のグレースケールのデータを 0 から 1 までの値に変換し、$8 \times 8 = 64$ 次元のデータを 256 個使って訓練した。

学習係数は 0.01 とし、MSE $= 0.03$ に達するまでに要した学習回数は 1460 回であった。

図 10: 恒等写像により復元された画像

2.5 実習：英単語の読み

Plaut ら (Plaut, McClelland, Seidenberg & Patterson, 1996) は単音節の英単語 2998 語を音韻に変換するモデルを提出している。彼らの用いたのは 3 層のバックプロパゲーションである。

Plaut らのモデルをトライアングルモデルという。トライアングルモデルにおける読みの説明は以下のとおりである。書記素層から音韻層への直接経路では、多くの単語と発音規則が一致する規則語と高頻度の不規則語が学習される。一方、低頻度の不規則語は意味系に依存すると仮定される。すなわちトライアングルモデルにおける直接経路では規則語と高頻度例外語が学習され、従って直接経路は単語の頻度効果に、すなわち単語の統計情報 (生起確率) に敏感である。規則語および高頻度例外語と低頻度例外語との処理の違いには労働の分割と呼ばれる作用が関与する。

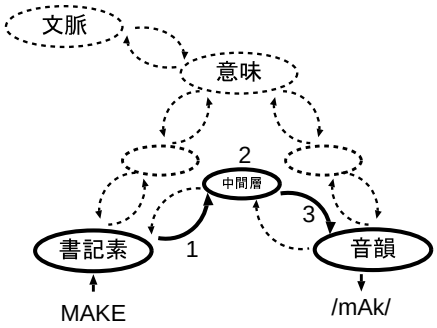

図 11: トライアングルモデル

単語の入出力表現にはオンセット、母音、コーダという表現を持ちている。単音節の単語なので、母音については一つだけのコーディングが必要である。加えて母音の前後に子音のクラスターが必要である。母音の前の子音をオンセット、母音の後の子音をコーダという。

まず、入力表現である orthography は、オンセットが Y, S, P, T, K, Q,

C, B, D, G, F, V, J, Z, L, M, N, R, W, H, CH, GH, GN, PH, PS, RH, TH, TS, WH の 30 とおりであり、母音は E, I, O, U, A¡ Y, AI, AU, AW, AY, EA, EE, EI, EU, EW, EY, IE, OA, OE, OI, OO, OU, OW, OY, OW, OY, UE, UI, UY の 27 とおり、コーダは H, R, L, M, N, B, D, G, C, X, F, V, J, S, Z, P, T, K, Q, BB, CH, CK, DD, DG, FF, GG, GH, GN, KS, LL, NG, NN, PH, PP, PS, RR, SH, SL, SS, TCH, TH, TS, TT, ZZ, U, E, ES, ED の 48 とおりであった。すなわち入力表現は計 105 次元のベクトルとして表現された。

出力表現である phonology の表現としては、オンセットが (s, S, C), (z, Z,j,f,v,T, D, p, b, t, d, k, g, m, n, h), (l, r, w, y) の 23 次元、母音が a, e, i, o, u, @, ^, A, E, I, O, U, W, Y の 14 次元、コーダが (r), (l), (m, n, N), (b, g, d), (ps, ks, ts), (s, z), (f, v, p, k), (t), (S, Z, T, D, C, j) の 24 次元の計 61 次元のベクトルとして表現された。出力表現は彼ら読字の表記方法であり、いわゆる発音記号とは関係がない。表記は次のようなものである。/a/ は POT の、/@/ は CAT の、/e/ は BED の、/i/ は HIT の、/o/ は DOG の、/u/ は GOOD の、/A/ は MAKE の、/E/ は KEEP の、/I/ は BIKE の、/O/ は HOPE の、/U/ は BOOT の、/W/ は NOW の、/Y/ は BOY の、/^/ は CUP の、/N/ は RING の、/S/ は SHE の、/C/ は CHIN の、/Z/ は BEIGE の、/T/ は THIN の、/D/ は THIS の音を各々表現している。

母音の前後にある子音には順序関係についての制約がある。例えばオンセットクラスターにおける /s/, /t/, /r/ は順序が /str/ でなければならない。出力表現である phonology のオンセットとコーダにあるカッコ内の音が相互に排他的な表現であることを意味している。この制約によって子音の順序が一意的に定まるように。なっている。子音は必ず上に表記した順序で音声化されるという制約がある。

これに加えて単語 CLASP と LASPE とでは /p/ と /s/ との順序関係を表現できないため /ps/ というユニットが加えられている。同じ理由により /ks/, /ts/ というユニットが加えられた。

英語はアルファベットを表記記号とする言語であるが、単語の書記形態の一部が音韻形態に対応しているに過ぎない。そこで orthography のユニットしては単一文字からなるユニットの他に 2 つの文字の組み合わせからなるユニットも用いられた。

これらの表現の詳細については Plaut ら (Plaut, McClelland, Seidenberg & Patterson, 1996) の原典を参照して欲しい。

3 層のバックプロパゲーションネットワークを用いて Pluat らの学習させた 2998 単音節単語を学習させた。中間層のユニット数は彼らのシミュレーションと同じ 100 個にした。MSE=0.0305 程度にまで学習が進行し、このときの正解率はおよそ 94.46 % であった。同じ学習セットをパーセプトロンで学習させる (MSE=0.05) と 87.26 % ほどの正解率になる。

ちなみに中間層のユニット数を 30 にしても学習が成立する。中間層のユニット数 30 のときの MSE=0.05 の場合正解率は 89.26 % であった。

学習の成立したネットワークを用いて、Plaut らの論文にあるような非単語を入力した結果が表 2 である。

表 2: Glushko(1979) の非単語を読ませた結果

	一貫語	非一貫語
人間	93.8	78.3
PMSP96	97.7	72.1
bp3(中間層 100,MSE=0.03)	90.7	53.5
bp3(中間層 100,MSE=0.05)	95.3	58.1
bp3(中間層 30,MSE=0.05)	88.4	58.1
perceptron(MSE=0.05)	93.0	67.4

人間の被検者が読んだ場合 93.8 % の正解率の一貫語が Plaut らのモデルでは 97.7 % と読めているのに対して、bp3 では 90.7 % であり、パーセプトロンでは 93.0 % であった。非一貫語についても bp3 が 53.5 % であるのに対してパーセプトロンでは 67.4 % であった。このことは訓練した 2998 単語の正解率ではパーセプトロンの成績は最も悪かったにもかかわらず、非単語の読みに対してはパーセプトロンは人間の読みの成績に近いものになっていることが分かる。このことは逆説的ではあるが、より能力の高いバックプロパゲーションを使うよりも、よりシンプルなパーセプトロンを使った方が、一見すると矛盾するような結果になっている。これは 2998 単語の読みを学習させるためにバックプロパゲーションによる学習では平均二乗誤差が 0.03 になるまで学習させた結果かも知れない。すなわち訓練のしすぎによる過学習が起こってしまったため、非単語の読みにおける一般的な能力が低下してしまったためかも知れない。平均二乗誤差を 0.05 で打ち切ると訓練データである 2998 単語の正解率は 88.4 % と落ちるものの glushko の非単語リストの成績は平均二乗誤差を 0.03 とした場合より向上した (表 2)。

このように学習をどこで打ち切るかと言う問題は注意を要するものである。また、Plaut らの結果とやや異なる結果が得られたことは興味深く、示唆に富んでいる。

3　バックプロパゲーション法の加速

3.1　モーメント法

実際の計算では、ϵ が小さいと学習回数が多くなってなかなか収束しない。そこで、式 (4) の代わりに、

$$\Delta \boldsymbol{w}_{ij}^{k-1,k}(t+1) = -\epsilon \frac{\partial E}{\partial \boldsymbol{w}_{ij}^{k-1,k}} + \alpha \Delta \boldsymbol{w}_{ij}^{k-1,k}(t), \tag{19}$$

のような修正を加えることが行われる。これは、1 つ前の修正量も考慮して結合係数を調整することを意味する。厳密には最急降下法ではなくなるが、学習が高速になる。式 (19) で前回の修正量と符号が異なると w_{ij} の更新量は小さな値になるため更新量が極端になるのを抑制する働きもある。

3.2　修正モーメント法

最急降下法では、極値付近の微分係数が小さいと収束が遅くなるという欠点がある。そこで、学習回数が増えるに従ってモーメントからの修正量を大きくして学習を促進してやる方法がある。

$$\Delta \boldsymbol{w}_{ij}^{k-1,k}(t+1) = -\epsilon \frac{\partial E}{\partial \boldsymbol{w}_{ij}^{k-1,k}} + m(t) \Delta \boldsymbol{w}_{ij}^{k-1,k}(t), \tag{20}$$

$$m(t+1) = m(t) + \Delta m \tag{21}$$

4　バックプロパゲーション法の問題点

以上見てきたバックプロパゲーション法であるが、実際にシミュレーションをするときには、幾つかの問題が発生する。学習が収束しない場合どうするのか、学習係数をどのくらいにするのがよいのか、どのくらいの誤差になったら学習を打ち切るのか、中間層の数を幾つにすれば良いのか、などである。また、一旦学習が成立したネットワークが本当に望んでいる知識を獲得したのか、を検証するためには訓練時とは別のデータを用いて一般性を検証しなければならない。これらの問題は、納得できるまでシミュレーションを繰り返し実行することで解決するいわゆる力業 (brute force method) でもよいのだが、数学的な解決策も研究されている、以下では詳細に立ち入ることはしないが要点だけを述べておく。

4.1　ローカルミニマム

局所最小 (ローカルミニマム local mimima) に陥って最適解に収束しないことがある。これは勾配降下法の持つ欠点である。例えば、(3) 式のような誤

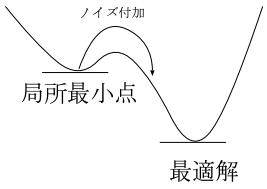

図 12: 局所最小の概念図

差関数が図 12 のようになっていたとすると、図中の 2 点で微分係数がゼロ (接線が水平) になる。このとき学習は進行しない。

ローカルミニマムから抜け出せるように、

$$\tau \frac{dw}{dt} = -\frac{\partial E}{\partial w} + \sigma W \tag{22}$$

ホワイトノイズ W を加えて、σ を時間とともにゼロに近づけることによって最適解を見つけ出せることが (Geman & Geman, 1984) によって証明されている。

4.2 係数爆発

中間層の係数が指数関数的に大きくなることがある。例えば、対称性を認識するネットワークの例で、結合係数は指数関数的に大きくなっていく場合がある。認識すべき範囲が大きくなるとどうなるかを考えるとこの問題は深刻である。

また、ネットワークの出力を厳密に 0 や 1 に近づけるとシグモイド関数の性質から、0 や 1 になるのはそれぞれ $-\infty, +\infty$ の場合であるため結合係数の絶対値が大きくなりやすい。実際の問題では 0 と 1 の代わりに、0.1 と 0.9 にするなどの工夫をする必要がある。

4.3 バッチ学習が良いのかそれともオンライン学習がよいのか

結合係数の更新方式として、1 エポックすべての結合係数の増減を保持しておき、1 エポック終了時に一度にその差分を更新するバッチ方式と、刺激が与えられるたびに、少しずつ結合係数を変化させるオンライン方式がある。どちらがよいとは一概に言えないが、学習させるデータの順番をランダムにするオンライン方式では局所最小を確率的に迂回できる可能性があって有利だと言われる。しかしオンライン方式では、最初に学習したパターンに後続の学習が影響を受けるので学習係数が大きいときには局所最小に陥りやすいという側面もある。どちらの学習方式が良いのかは解くべき問題にもより、一般的な結論を出すことはできない。

5　一般化能力と過学習

学習すべきデータ集合の中からいくつかのサンプルを選び訓練課題セットとしてニューラルネットワークに学習させるとする。このとき、学習させた学習課題セット以外のデータをテスト課題として選び、このテスト課題の成績を調べることで学習したルールの一般化能力を測定することができる。

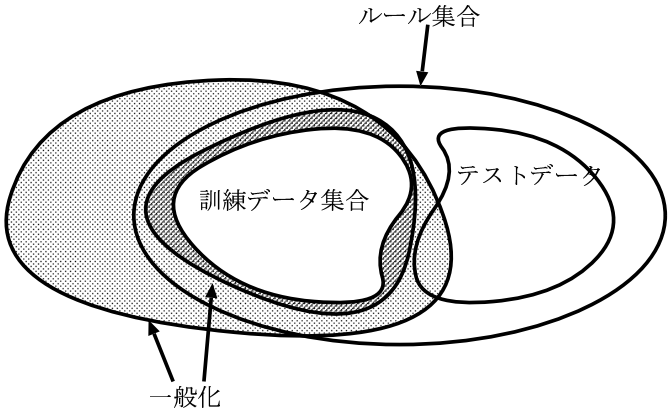

図 13: 一般化能力と過学習

学習した内容がルールの適用範囲と完全に重なることが理想であるが、図 13 で示したように一般化にはさまざまな可能性が考えらる。例えば、入出力とも 0, 1 のデータで、入力層が N 個、出力層が 1 個である場合を考えてみよう。入力パターンの総数は N 個の入力層の可能な組み合わせ、すなわち 2^N 個存在します。これらの入力集合を 0 か 1 かに分類する課題では、全部で 2^{2^N} とおりの分類が可能である。この中から M 個の入力を選んで訓練した場合には、残りの分類パターンはすべて一般化になるので $2^{2^N} - M$ とおりの一般化が可能になります。このことは入力層のユニット数 N が大きくなると、実質的に無限大の一般化が考えられることを意味している。

一方、訓練課題セットでは正解を得ることができるが、テスト課題では正解できないことがある。学習が進行しすぎるとしばしば観察される現象で、過剰な学習がなされたことを意味する (overfitting 過学習または過剰適合と呼ばれる)。図 13 は、誤った方向に一般化がなされた場合と、過剰適合によって訓練課題にだけ正解するようになった場合とを表したものである。

図 14 では正弦曲線 $y = 0.5 + 0.4 \sin(2\pi x)$ を 3 層のネットワークに学習させた例を示した。この例では 0 から 1 までの 0.05 刻みの各 x 座標を入力信号として与え、対応する y の値を学習させた。実際の教師信号に若干のノイズを加えてある。一般に教師信号に少量のノイズを加えたデータを学習させることで、ネットワーク一般化能力が向上すると言われている。ただし、デー

タセット数に対して中間層の数が多いときに繰り返し学習を進行させると過剰適合が生じることがある。図 14 中の 2 本の点線のうち、ノイズを付加し

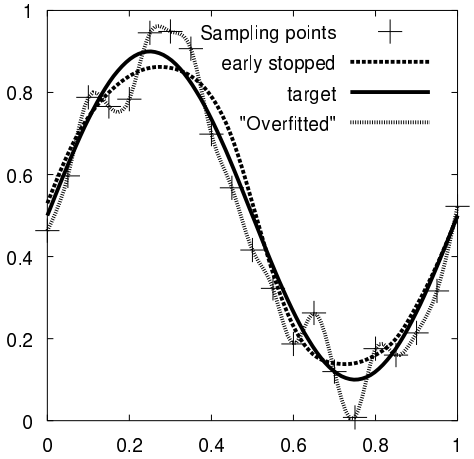

図 14: 正弦曲線を 3 層バックプロパゲーション法によって学習させた例。実線が $0.5 + 0.4\sin(2\pi x)$ の曲線であり、プラス (+) の記号で実際に用いられた教師信号 (ノイズ付加) が示されている。2 本の点線によって、中間層を少なくして学習終了基準を甘く設定した結果と、中間層の数を増やして意図的に過剰適合を起こさせた結果とを示した。

た教師信号を完全に学習している点線では、過学習のために真の曲線を学習するのではなく真の関数とノイズとの合成積 convolution を学習してしまっている。他方の点線では、ほぼ望み通りの結果を得ているが[1]、最大値 0.9 付近、最小値 0.1 付近での真の関数とのずれがやや大きくなっていることが読み取れる。

6 中間層ユニット数の決定問題

一般化能力と過学習の節でも言及したが、一般に、中間層のユニット数が多すぎると過学習を起こしやすく、反対に中間層のユニット数が少なすぎると学習が成立しないといわれる (Bishop, 1995)。

入力情報が N ビットの情報を持っているとき、 全入力情報を損失無く表現するためには 2^N 個の中間層ユニットを用意すれば十分であることはすぐに分かる。ところが、これは中間層のユニット数決定のための必要条件ではない。では最適な中間層のユニット数は幾つなのだろうか？最適な中間層のユニット数を決定するためには、5 節の一般化の問題を踏まえて議論する必

[1] 実際に用いた中間のユニット数は 6

要がある (甘利、村田、Müller(1997), Elman ら (1996))。

村田ら (1994) は、神経回路網を確率機械と捉えて、情報量規準を用いて中間層を定める手法を提案した。例えば、神経回路網を確率機械と捉えて、情報量規準を用いて中間層を定める手法が村田ら (Murata, Yoshizawa & Amari, 1994) によって提案されている。彼らは赤池の情報量規準 AIC (Akaike's Information Criterion, 坂本、石黒 & 北川 (1983)) を拡張した NIC (Network Information Criterion) を提案した。データが与えられたときのモデルの対数尤度に自由パラメータ数の 2 倍を加えたものが赤池の情報量規準 AIC と呼ばれる量である。(データの当てはまりを表す量である) 対数尤度が同じならば自由パラメータ数の少ないモデルを選択すべきであることを AIC は主張している。複数のモデル間で AIC を計算、比較して最適モデルを選択しようとするのが AIC の基本となるアイデアである。村田らの提案した NIC は、中間層のユニットをパラメータと考え、訓練データ上で、任意の入力信号に対するモデルからの出力と教師信号 (正解) との「ずれ」に、パラメータ数を加えたものとして定義されている。もし、モデルが真の入出力関係を実現可能であり、かつ、上記の「ずれ」が対数のマイナスで定義されているならば、NIC と AIC とは係数を除いて一致する。

6.1 実習：中間層の素子数

XOR 問題を解くネットワークで中間層ニューロンの結合係数を 0 にすることによって脳損傷を表現せよ。一中間層のニューロン数が 4,6,8 個のときに、一つの中間層のニューロンが損傷を受けると成績はどのようになるか。

損傷を受けるニューロンのうちで成績に影響がでやすいものと出にくいものとがある違いは何か? このような中間層のニューロンの性格はどのように評価したらよいか考えよ。

4, 6, 8 個の中間層の中の 1 個の素子が損傷を受けたあとで再学習が可能かどうかプログラムを実行して誤差の減少の様子をグラフ化せよ。誤差の減少にパターンが見られるかどうかを検討せよ。

7 標準正則化理論と weight decay, weight elimination

一般化と過学習、さらに中間層のユニット数問題とも関連する技法として正則化項を導入する一連の手法がある。正則化項を導入することでネットワークの一般化能力を高めたり、最適なネットワークを探しだす方法である。この手法は、初期視覚過程における計算理論である標準正則化理論とも関連付けられるもので、最適化すべき目標関数に正則化項を加えた関数を最適化することによって定式化される (Poggio, Torre & Koch, 1985)。

標準正則化理論に基づく手法では誤差関数に滑らかさの制約を示す項加えたものを新たな誤差関数として定義し、この関数を最小化するようにネットワークを訓練する。

$$\tilde{E} = E + \nu\Omega \tag{23}$$

ここで E はこれまで用いてきた誤差関数である。パラメータ ν はペナルティ項 (正則化項) Ω の影響を支配する定数である。ν が大きければ正則化項の影響が大きくなり、小さければ正則化項の影響を受けないにくくなる。

7.1 weight decay

最も単純な方法は weight decay と呼ばれもので、正則化項として

$$\Omega = \frac{1}{2}\sum_i w_i^2 \tag{24}$$

を用いる。weight decay には結合係数が不必要に大きくなるのを抑制する効果がある。すなわち、結合係数の爆発 (4.2 節) を抑えることによって過学習を抑止するという効果が期待できる。

具体的には w_{ij} の更新後に

$$w_{ij}^{\mathrm{new}} = (1-\epsilon)w_{ij}^{\mathrm{old}} \tag{25}$$

を用いる。これはバックプロパゲーションで使われる誤差の 2 乗和 E に w_{ij} の 2 乗和を加えた

$$\widetilde{E} = E + \frac{\nu}{2}\sum w_{ij}^2 \tag{26}$$

を新たに誤差関数として誤差逆伝播アルゴリズムによる結合係数の更新式

$$\Delta w_{ij} = -\eta\frac{\partial\widetilde{E}}{\partial w_{ij}} \tag{27}$$

を適用するのことと同じである。2 乗誤差 E が 0 であるとみなせるとき、上式を連続近似できると考えて

$$\frac{dw_{ij}}{d\tau} = -\eta\nu w_{ij} \tag{28}$$

とすれば、この微分方程式の解は

$$w_{ij}(\tau) = w_{ij}(0)\exp\left(-\eta\nu\tau\right) \tag{29}$$

という指数関数になって 0 に漸近するからである。

weight decay は枝刈り法 pruning すなわち、最適なネットワーク構造を探索するための手段として用いられることがある。最初は全結合を作っておいて、正則化項を用いて学習を行ない、ある一定のしきい値以下の結合係数を

ゼロとしてネットワークを簡略化する方法である。最初に全結合を作っておいて、あとで不要なものを刈り取ることは、網膜から外側膝状体への投射、外側膝状体から第一次視覚野の間でも観測されている事実である。脳内でどのユニットがどの役割を果たすかが出生直後から決まっているわけではない。出生後の環境によって柔軟に対応できるようにするためには、このほうが有利なのだとの解釈も成り立つ。

7.2 weight elimination

weight decay は二乗誤差の減少に貢献していない結合係数の大きさを減じる傾向にある。枝刈り法との関連でいえば、wegiht decay は大きな結合係数よりも小さな結合係数を好む手法であるといえる。一方、以下の

$$\widetilde{E} = E + \nu \sum_i \frac{w_i^2}{\widehat{w}^2 + w_i^2} \tag{30}$$

で与えられる式を正則化項として用いる手法を weight elimination という。正則化項を w_i で微分すれば

$$\frac{\partial}{\partial w_i} \sum_i \frac{w_i^2}{\widehat{w}^2 + w_i^2} = \frac{2w_i(\widehat{w}^2 + w_i^2) - 2w_i^2 w_i}{\left(\widehat{w}^2 + w_i^2\right)^2} = \frac{2w_i \widehat{w}^2}{\left(\widehat{w}^2 + w_i^2\right)^2} \tag{31}$$

となる。この式は、w_i が $\widehat{w}$ に比べて十分に小さければ、分母が定数と見なせることになり、weight decay と同じように 0 に近づくことが分かる。一方 w_i が $\widehat{w}$ に比べて大きければ大きい程 $1/w^3$ のオーダーで変化が小さくなる。すなわちこの正則化項は、小さな結合係数を 0 に近づけ、大きな結合係数はそのままにするという傾向があることが分かる。この性質により weight elimination では、小さな結合係数と大きな結合係数が残り、中間の結合係数が減少する、めりはりの利いたネットワークを構成しやすいことを意味する。

文献

坂本, 石黒 & 北川 (1983). **情報量統計学**. 東京: 共立出版.

甘利俊一, 村田昇 & Muller, R. (1997). 学習の数理モデル—汎化能力と過学習—. In 外山敬介 & 杉江昇 (Eds.), **脳と計算論** chapter 3, (pp. 37–53). 東京: 朝倉書店.

Bishop, C. (1995). *Neural Networks for Pattern Recongnition*. Oxford University Press.

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Simth, A., Parisi, D. & Plunkett, K. (1996). *Rethinking Innateness: A connnectionist*

perspective on development. Cambridge, MA: MIT Press. (邦訳「認知発達と生得性」, 乾, 今井, 山下訳, 共立出版).

Geman, S. & Geman, D. (1984). Stochstic relaxation, gibbs distributions, and the bayesian restorion of image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6*, 721–741.

Murata, N., Yoshizawa, S. & Amari, S. (1994). Network information criterion — determining the number of hidden units for an artificial neural netwrork model. *IEEE Transactions on Neural Networks*, *5(6)*, 865–872.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S. & Patterson, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, *103*, 56–115.

Poggio, T., Torre, V. & Koch, C. (1985). Computational vision and regularization theory. *Nature*, *317*, 314–319.