数理科学特論Ａ～画像数学～　第１０回

シリーズ４：ニューラルネットワークと最適化問題

(1) ニューラルネットワークと誤差逆伝播学習

「画像数学」の第４シリーズは, ニューラルネットワークをとりあげます. なぜ「画像数学」にニューラルネットが関係あるのかと思われるかもしれませんが，実は視覚からのパターン認識の研究はニューラルネットの研究で重要な地位を占めているため, 画像処理とニューラルネットには密接な関係があります．このシリーズでは，ニューラルネットの基礎と学習アルゴリズム，画像処理とニューラルネットワークの関係，さらにニューラルネットで重要な意味をもつ「最適化」を確率的に行うシミュレーティッド・アニーリングと遺伝的アルゴリズム，の３つについて説明します.

ニューロンとニューラルネットワーク

ニューラルネットの研究は，まず神経素子すなわち**ニューロン**(neuron)をモデル化することから始まりました. 生理学的研究から,「ニューロンは他のニューロンから電気的・化学的刺激を受け，その刺激の総和が一定量を超えると「発火」して他のニューロンに刺激を与える」というモデルが提示されました．これを図１のような記号で表します．単純化するため，i番のニューロンのx_iの状態(status)を$x_i = 1$→発火している，$x_i = 0$→発火していないとし，発火しているとき１の量の刺激が他のニューロンへ送られます．また，刺激を受け取る際，神経素子間の結合には重みが設定されており，ひとつのニューロンには各結合から（重み×１ or ０）の刺激が届きます．すなわち，j番のニューロンの状態をx_jとし，j番とi番のニューロン間の結合重みをw_{ji}とすると，j番のニューロンからi番のニューロンに届く刺激の量は$w_{ji}x_j$となります．そして，その総和$\sum_j w_{ji}x_j$が一定量（しきい値(threshold)）を超えるとこのi番のニューロンが発火します．すなわち，しきい値をTとして，しきい値関数fを

$$f(x) = \begin{cases} 1 & x \geq T \\ 0 & x < T \end{cases} \qquad (1)$$

とするとき, $f(\sum_j w_{ji}x_j) = 1$ ならばi番のニューロンが発火，すなわち$x_i = 1$ とし，それ以外は$x_i = 0$ とします.

ニューラルネットによる情報処理では，このようなニューロンを多数結合し，ニューロンの初期状態を与えることで情報を入力し, 上のような刺激の伝達によってニューロンの状態を変化させて変化後のニューロンの状態を出力とします．ニューロンの結合の形（トポロジー）には，おおまかにいって**階層型**と**相互結合型**の２種類があります.

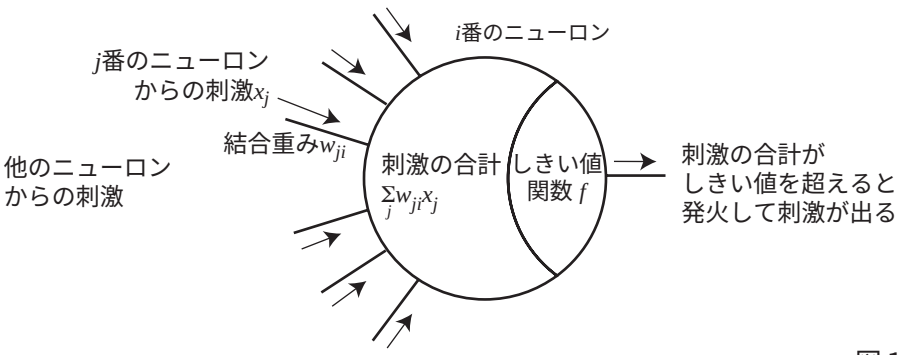

図１．ニューロンのモデル

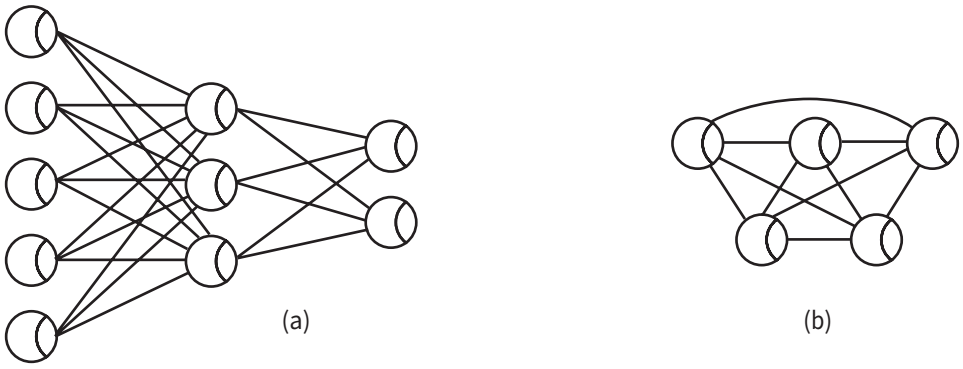

図２．ニューラルネットワークのトポロジー．(a) 階層型 (b)相互結合型

図２の(a)が階層型ネットワークで，視神経における情報処理をモデルとして生まれたものです．１つの層にあるニューロン全体で１つの情報を保持し（例えば，ニューロンを画素と考えて視覚的パターンとする），これがある層から次の層へ一定方向に刺激となって流れてゆきます．そして，最終層に刺激が届いたときの最終層のニューロンの状態が，このニューラルネットワークの出力結果となります．階層型ネットワークは，画像処理やパターン認識に多く用いられます．

一方，図２の(b)が相互結合型ネットワークで，こちらは層はなく各ニューロンが互いに各ニューロンと結合しています．こちらの場合は，結合重みを一種の「プログラム」として問題に合わせて適切に設定し，ニューロンの初期状態を設定した後互いに刺激を交換しあいます．その動作が収束してそれ以上刺激のやりとりが起こらなくなったとき，その時のニューロンの状態を出力結果（あるいは問題の解）とします．相互結合型ネットワークは，最適配置問題を解いたり，連想記憶（記憶したいニューロンの状態に応じて結合重みを設定しておき，ニューロンの初期状態を与えて刺激を交換しあうと，記憶した状態の中で初期状態にもっとも近い状態に収束する．すなわち，初期状態からそれにもっとも近い記憶が「想起」される）として用いられます．

以下，この講義では，画像処理に関連の深い階層型ニューラルネットワークのみを取り扱います．

パーセプトロン学習とその限界

パーセプトロン(perceptron)とは，ニューラルネットワーク研究の初期に提案された，もっとも簡単な階層型ニューラルネットワークで，図３のように入力層，中間層，出力層の３層からなっています．その利用法は，例えば入力層に「A」という文字パターンが入力されると出力層で「A」に対応するニューロンが発火し，以下，「B」のパターンにはBのニューロンが，「C」のパターンにはCのニューロンが，... というようにしてパターン認識を行うというものです．

パーセプトロンをはじめとする階層型ニューラルネットワークの特徴は，結合重みを**学習**によって定めることです．学習とは，ニューラルネットワークが望みの出力をするようにするための結合重みの調整を，入出力の例のみを用いて行うことです．パーセプトロンにおいては，学習は中間層と出力層の間だけで，以下のような手順で実現されます．以下，簡単のため出力層のニューロンは１つだけでその状態を x^O とします（図４）．

１．所望の入出力の例をいくつか用意します．これは例えば，入力層に「A」というパターンを入力したとき出力層のニューロンの状態が「1」になってほしい，といったものです．各入出力例において，入力層に入力例を与えたときの，中間層の j 番目のニューロンの状態を x_j^H とします．

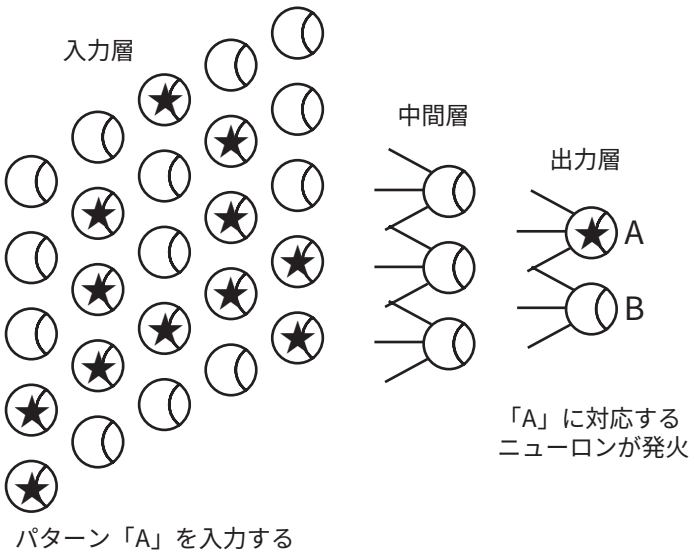

図３．階層型ニューラルネットワークによるパターン認識

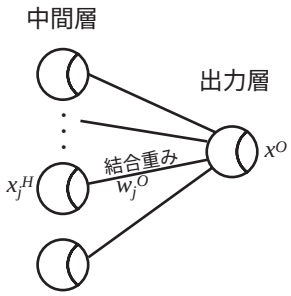

図４．２層間の結合重みの学習

２．中間層と出力層間の初期結合重み w_j^O を適当に（例えばランダムに）決めておきます.

３．入力の例を入力層に与え，その時の出力 x^O を調べます.

４．出力層での所望の出力を y^O とします．もし，所望の出力が $y^O = 1$ なのに現実の出力が $x^O = 0$ である（$x^O - y^O = -1$)ならば，それは出力層のニューロンにやって来る刺激の総和 $\sum_j w_j^O x_j^H$ が足りないからです．そこで，中間層のうち状態 x_j^H が 1 であるようなニューロンとの結合重み w_j^O を増やします．また，もし，所望の出力が $y^O = 0$ なのに現実の出力が $x^O = 1$ である（$x^O - y^O = +1$)ならば，それは出力層のニューロンにやって来る刺激の総和 $\sum_j w_j^O x_j^H$ が多すぎるからなので，中間層のうち状態が 1 であるようなニューロンとの結合重みを減らします.

以上のことは, 中間層の j 番のニューロンと出力層のニューロンとの結合の新しい重み w'^O_j を

$$w'^O_j = w_j^O - (x^O - y^O)x_j^H \tag{2}$$

にしたがって更新することで実現できます.

５．所望の出力が得られるようになるまで，３，４を繰り返します.

この方法をパーセプトロン学習といいます．この方法は簡単ですが, 学習アルゴリズムが収束せずに結合重み w_j^O がいくつかの組合せの間を循環してしまう場合があることが知られています. これは，1 回の学習で w_j^O から w'^O_j へ修正する修正量が大きすぎ, 急激に結合重みを修正しすぎるからです. そこで，しきい関数 f を例えば次のシグモイド関数

$$f(x) = \frac{1}{1 + \exp(-\lambda(x - T))} \tag{3}$$

のような連続関数に置き換えます（図５，λ はパラメータで，$\lambda \to \infty$ のとき(3)式の f はしきい関数になります）．そして，ニューロンの状態 x_j^H や x^O を連続値とし, (2)式に小さな正の実数 ε（学習係数といいます）を導入して

$$w'^O_j = w_j^O + \varepsilon(y^O - x^O)x_j^H \tag{4}$$

のように「徐々に」学習する方法があります．これは δ– ルールとよばれています.

さて，パーセプトロンでは出力層のニューロンの状態が(1)式で述べたとおり

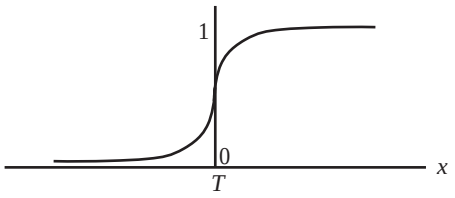

図５．シグモイド関数

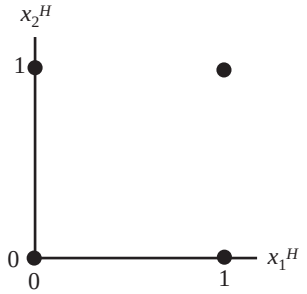

図６．ニューロンの状態．(x_1^H, x_2^H)の組は図の４つの●のいずれかにある．

$$x^O = f(\sum_j w_j^O x_j^H) \tag{5}$$

となります．この操作は，中間層のニューロンの状態の重み付き線形和を求め，それがしきい値Tより大きいか小さいかで出力層のニューロンの状態を決めることを意味しています．

そこで，中間層のニューロンがx_1^H, x_2^Hの２つしかない場合を考え，両ニューロンの状態を図６の座標平面で表します．このとき，(5)式の計算は図７(a)の座標平面上で

$$w_1^O x_1^H + w_2^O x_2^H = T \tag{6}$$

という直線をひき，現在の中間層のニューロンの状態x^H_1, x^H_2がこの直線のどちらにあるかによってx^Oが１か０かを決定していることになります．

ところが，例えば「x^O=［x_1^Hとx_2^Hの排他的論理和（XOR)］」という演算では，図７(b)のように，x^Oが１になるようなx_1^Hとx_2^Hの値の領域とx^Oが０になるようなx_1^Hとx_2^Hの値の領域とを直線で区切ることができません（このような演算を**線形非分離**といいます)．したがって，どのように学習しても中間層と出力層の２層では［x_1^Hとx_2^Hの排他的論理和］という演算を得ることはできません*)．パーセプトロンでは入力層と中間層との間の結合重みは固定されていますから，入力層と中間層との間で行う演算の設定によっては，どう学習しても求めるパターン認識ができない，ということが起こります．

これは，パーセプトロンの限界の１つです．これを指摘したMinskyとPapertは，さらに入力層と中間層の間の構成に現実的な制限を課すと表現できない問題がたくさんあることを述べ，中間層と出力層の間の学習だけでは大したことは学習できないことを示しました．その結果，ニューラルネットワークの研究は一気に下火になってしまったのです．この限界を超えるには，中間層と出力層間だけでなく，ネットワークの層間の結合全体を学習する方法を考える必要があります．それを実現したのが，次章で述べる誤差逆伝播法です．

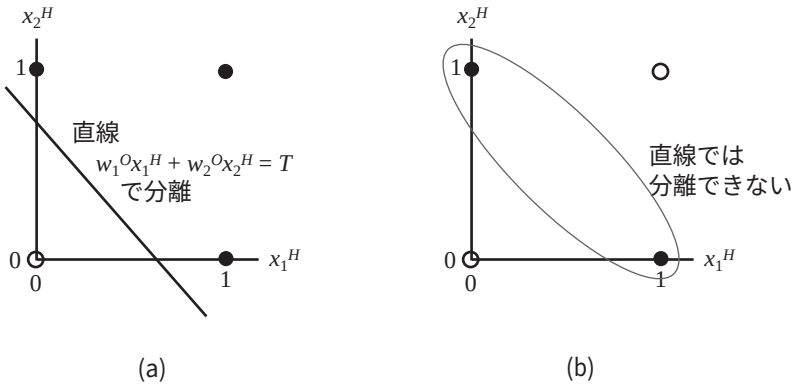

図７．線形分離性．(a) ●と○は線形分離．(b) 排他的論理和．●と○は線形非分離．

*)「パーセプトロン学習が収束しない場合」とは，所望の入出力例が線形非分離であるときです．δ–ルールでは，この場合入出力例との誤差の２乗が最小である結合重みに収束します．

誤差逆伝播法

前節のパーセプトロン学習やδ–ルールにおいては,「所望の入出力例」は入力層と出力層の状態としてしか与えられていません．ですから，学習によって最小化する「入力例に対する，所望の出力と現実の出力との誤差」は，出力層についてしか得られません．パーセプトロン学習やδ–ルールにおいては，この誤差を使って出力層とその１つ手前の層（全体で n 層あるとすれば，第$(n-1)$層）との結合重みを調整しました．

この方法を使ってさらにもう１つ手前の層（第$(n-2)$層）との結合重みを調整するには，第$(n-1)$層の各ニューロンでの誤差を知る必要があります．そのためには，出力層の１つのニューロンでの誤差を第$(n-1)$層の各ニューロンに割り振らなければなりません．この割り振りを，次の考え方で行います．

[1] 第$(n-1)$層の各ニューロンのうち，出力層のこのニューロンとの結合重みが大きいニューロンにより多く誤差を割り振るほうが，割り振られた誤差が修正されたとき，その修正の出力層での効果が大きくなって好都合です．

[2] 第$(n-1)$層の各ニューロンの状態は，第$(n-2)$層から送られてくる刺激の総和にシグモイド関数のような非線形関数を適用して決まります．このとき，刺激の総和の変化に対するニューロンの状態の変化率が大きい，つまり「感度のよい」ニューロンにより多く誤差を割り振るほうが，やはり割り振られた誤差の修正の効果が大きくなって好都合です．

これを，一般の第 k 層と第 $k+1$ 層について，記号と式を使って書くと，つぎのようになります（図８を参照してください）．

いま，第 k 層の j 番目のニューロンが受け持つ誤差の量を δ_j^k とし，第 $k+1$ 層の l 番目のニューロンが受け持っている誤差の量を δ_l^{k+1} とします．また，第 k 層の j 番目のニューロンから $k+1$ 層の l 番目のニューロンへの結合の重みを $w_{jl}^{k(k+1)}$で表します．さらに，第 k 層の j 番目のニューロンの状態を x_j^k で表します．

さて，上の「考え方」[1]の条件は「δ_j^k のうち第 $k+1$ 層の l 番目のニューロンから割り振られる量」が $w_{jl}^{k(k+1)}$に比例することを意味します．また，「第 k 層の j 番目のニューロンに第 $k-1$ 層から送られてくる刺激の総和」S_j^k は

$$S_j^k = \sum_i w_{ij}^{(k-1)k} x_i^{k-1} \qquad (7)$$

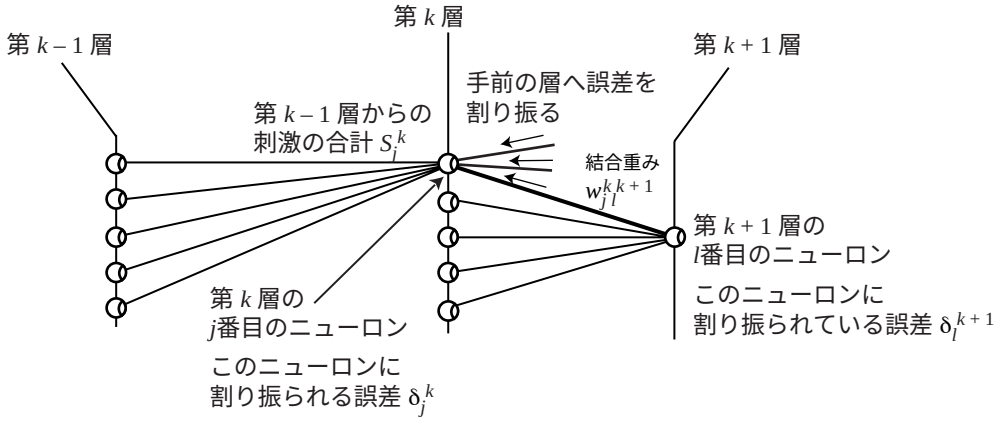

図８．誤差逆伝播法．

と表せますから, [2]の条件は,「δ_j^k のうち第 $k+1$ 層の l 番目のニューロンから割り振られる量」が dx_j^k / dS_j^k にも比例することを意味します. これらの誤差が第$k+1$層の各ニューロンから割り振られて, その合計が δ_j^k となるわけですから

$$\delta_j^k = \sum_l \left(w_{jl}^{k\,k+1} \frac{dx_j^k}{dS_j^k} \right) \delta_l^{k+1} \tag{8}$$

となります．ここで,（しきい関数の代用のシグモイド関数などの）非線形関数を $f(\ )$とすると

$$x_j^k = f(S_j^k) \tag{9}$$

ですから,

$$\frac{dx_j^k}{dS_j^k} = f'(S_j^k) \tag{10}$$

となり, (8)式は

$$\begin{aligned} \delta_j^k &= \sum_l \left(w_{jl}^{k\,k+1} f'(S_j^k) \right) \delta_l^{k+1} \\ &= \left[\sum_l \left(w_{jl}^{k\,k+1} \delta_l^{k+1} \right) \right] f'(S_j^k) \end{aligned} \tag{11}$$

と表されます．この式によって，第k層のj番目のニューロンが受け持つ誤差の量が第k層の各ニューロンが持つ誤差から決まります．$k=n$のとき，すなわち出力層については第$k+1$層がありませんが，出力層での誤差とは所望の出力と現実の出力との差ですから，出力層の l 番目のニューロンにおける所望の出力をy_j とすると

$$\delta_l^n = (x_l^n - y_l) f'(S_l^n) \tag{12}$$

となります.

(11)(12)式で，各層の各ニューロンが受け持つ誤差が決まりました．そこで，これらの誤差に(4)式の δ– ルールを適用して，新しい結合重み$w'^{k-1\,k}_{jl}$ を

$$w'^{k-1\,k}_{jl} = w_{jl}^{k-1\,k} - \varepsilon \delta_l^k x_j^{k-1} \tag{13}$$

のように結合の重みを修正します．ε は(4)式と同じ学習係数です.

以上の方法で３層以上の階層型ニューラルネットワークの学習を行う方法を，**誤差逆伝播法** (error back propagation method, EBP) といいます．この名前は，出力層での所望の出力と現実の出力との差をもとに，手前の層に誤差を順に伝えていくことから来ています.

さて，ここまでの説明では誤差逆伝播法を「考え方」[1][2]から導きましたが，実はこのようにして導かれた誤差に基づいた(13)式の結合重みの修正法は，最終層での誤差の２乗和がその時点でもっとも多く減る方向に結合重みを修正することが証明できます．このことを，この修正法が**最急降下** (steepest descent) になっているといいます．最急降下であることの証明は付録（ネット上にあります）に回すとして，ここでは最急降下とはどういう意味かを説明します.

結合重みのある組合せに対して，誤差２乗和は１通りに定まります．したがって，各結合重みを座標軸とする多次元空間の各点に，誤差２乗和の値が決まっているような「場」を考えることができます．結合重みの修正は，誤差２乗和がもっとも小さい点をめざして，多次元空間の中のある経路をた

どってゆくことに相当します．最急降下とは，現在いる点から誤差２乗和がもっとも大きく減少する方向へ進むことを意味しています．

これがどのような方向かを求めてみましょう．$\boldsymbol{x}$を多次元空間の一点とし，$f(\boldsymbol{x})$を各点の関数（上の例なら誤差２乗和）とします．ここで，sをパラメータとして$\boldsymbol{x}(s)$で多次元空間中のある経路を表すことにします．

このとき，経路上の$\boldsymbol{x}(s)$における勾配を表すベクトルは

$$\boldsymbol{b} = \frac{d\boldsymbol{x}}{ds} \tag{14}$$

で表されます．そこで，この経路の方向にfを微分すると，

$$\begin{aligned} \frac{\partial f}{\partial s} &= \frac{\partial f}{\partial \boldsymbol{x}} \cdot \frac{d\boldsymbol{x}}{ds} \\ &= \boldsymbol{b} \cdot \mathrm{grad}\ f \end{aligned} \tag{15}$$

が得られます．grad fは$f(\boldsymbol{x})$を各座標軸の方向に微分したベクトルで，たとえば$\boldsymbol{x} = (x, y, z)$という３次元空間なら，grad fは

$$\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right) \tag{16}$$

というベクトルになります．(15)式によると，経路上の勾配$\boldsymbol{b}$がgrad fに平行で逆方向であるとき，fの変化率が負で絶対値が最大になります．つまり，もっともfを減少させる方向，すなわち最急降下の方向は– grad fであることがわかります．

最急降下の方向は，fが現時点でもっとも減少する方向を示しているだけで，最小値に向かう方向を示しているのではないことに注意してください．この問題は，このシリーズの第３回でシミュレーティッドアニーリング・遺伝的アルゴリズムを説明するときに取り扱います．

今日の参考文献

麻生英樹，ニューラルネットワーク情報処理，産業図書(1988)．ISBN4-7828-5124-3
浅野　晃，博士学位論文，大阪大学(1992)．