数理科学特論Ａ～画像数学～　第１２回

シリーズ４：ニューラルネットワークと最適化問題

(3) 確率的最適化法

前々回に説明した階層型ニューラルネットワークは, 理想とする出力と現実の出力との誤差を最急降下法によって減少させることにより，ネットワークの最適化を行っています．しかし，最急降下法では，今回述べる「局所解」の問題のために，誤差を最小にする結合重み，すなわち最適解が得られないことがあります．この問題を解決する方法として提案されているのが，今回紹介するシミュレーティッド・アニーリングと遺伝アルゴリズムに代表される確率的最適化法です．最急降下法が「ただひたすら誤差を減らす」という方法であるのに対して，この２つの方法は「たまには誤差を増やすことがあってもいいじゃない」という「多様性」をもつことで，最終的によい解を高い確率で得ようとするもので，実社会とちょっと似たところがあります．

最急降下法の問題

前々回の講義で, 階層型ニューラルネットワークの学習を説明しました．階層型ニューラルネットワークの学習とは, 各層間の結合重みを徐々に変化させて, ネットワークの入力層に与えられるある入力に対して出力層で得られる出力と, その入力に対する理想的出力との誤差が最小になるようにすることです．前々回の講義で説明した誤差逆伝播法は，誤差についての最急降下法になっています．最急降下法とは，常に誤差がもっとも大きく減少する方向に各結合重みを変化させることです．

これをさらに一般的に言うと，

> いくつかのパラメータで入出力の関係が調整できる系があるとき, ある入力に対して現状の出力と理想の出力との誤差がもっとも大きく減少する方向に，パラメータを調整する

ということになります．このような考えで，とりうるパラメータの組み合わせが多すぎて，パラメータの全ての組み合わせを探索できない問題でも，適切なパラメータを求めることができます．

しかし，最急降下法には次のような問題があります．

[1]「誤差をもっとも大きく減少させる」ことと,「誤差の最小値に向かう」こととは意味が異なります．それは，図１の簡単な例でもわかります．図１は重みが１つしかない場合です．A 点から出発して常に「誤差をもっとも大きく減少させる」ように重みを変化させれば誤差の最小値に達しますが, B 点からでは最小値に達することが出来ず，L 点に達した時点でこれ以上重みは変化させられなくなります．このような L 点を局所解 (local minimum) といいます．このように，誤差の真の最小（global minimum)を与える最適解に達することができるかどうかは，出発点すなわち初期値に依存します．

[2] 誤差逆伝播法では, 誤差をもっとも大きく減少させるような重みの変化のさせ方を, 解析的に求めることができました．しかし，このようなことがどんな問題でも可能であるとは限りません．誤差逆伝播法では,「パラメータの変化に対する効果の変化率」，すなわち「刺激の重みつき合計の，重みによる微分」が求められることを利用して，最急降下になる重みの変化法を導きました．しかし，例えば「たくさんある要素部品から最適な組合せを求める」というような問題では，ふつうこのような「パラメータの変化に対する効果の変化率」は求められません．

今回の講義で説明するシミュレーティッド・アニーリング (simulated annealing) と遺伝的アルゴリズ

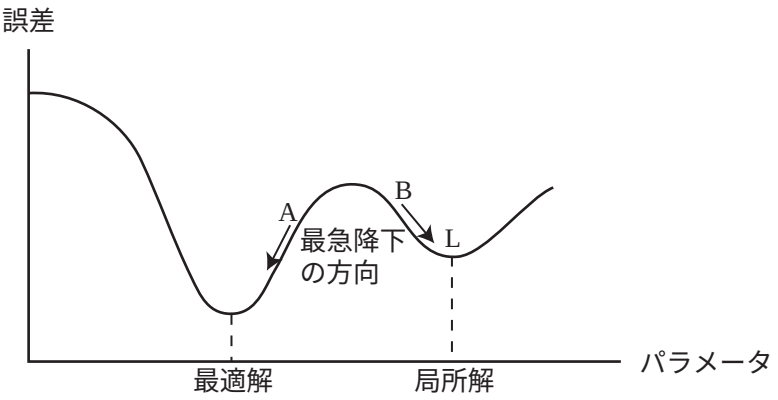

図１．
最急降下と局所解

ム (genetic algorithm) は，これらの問題にある意味での解決策を与えるものです．「ある意味」とは，「最適解が**確率的**に求まる」，すなわち，「最適解が確実に求まるわけではないが，最適解に達することのできる確率を高くする」という考え方です．

両者には共通する考え方があります．それは，

> とりあえず，パラメータを変化させてみて，その変化を受け入れるかどうかを誤差の変化によって決定する

という試行錯誤的な方法をとることと，

> 「常に誤差を減らす方向にパラメータを変化させるのではなく，誤差が増える方向へも進んでみる」ことで，局所解に落ち着いてしまうのを防ぐ

というものです．以下，**確率的最適化法**とよばれているこれらの両手法を説明します．

シミュレーティッド・アニーリング

「アニーリング」とは金属加工でいう「焼きなまし」で，金属を熱して加工した後徐々に冷やすことで金属材料の粘り強さを高める方法です．ですからシミュレーティッド・アニーリングは「疑似焼きなまし」と訳されることもあります．

この方法の本質は，

> 誤差が増える方向へのパラメータの変化も，**ある確率で**受け入れる

ことにあります．図２のように，こうすることで局所解から抜け出せる可能性が出てきます．

この確率は，まず「パラメータの変化による誤差の増加が小さいほど，大きく」なります．つまり，パラメータの変化によって誤差が少々増える程度なら，そのような変化も受け入れてみようというわけです．

さらに，この確率は最適化のはじめのうちは大きく，最適化が進むに連れて小さくしてゆきます．最適化のはじめはどこに最適解があるかわからないので，なるべくいろいろなパラメータを探索できるようにしておき，後半は解に近づいているところから抜け出してしまわないように確率を小さくします．この確率を決める値を，「最適化のはじめは温度が高いので活発に振動し，後半はだんだん冷めてきてあまり振動しなくなる」という類推で**温度**とよびます．「焼きなまし」という名前もここから来ています．

シミュレーティッド・アニーリングの手続きは概ね以下のようになります．

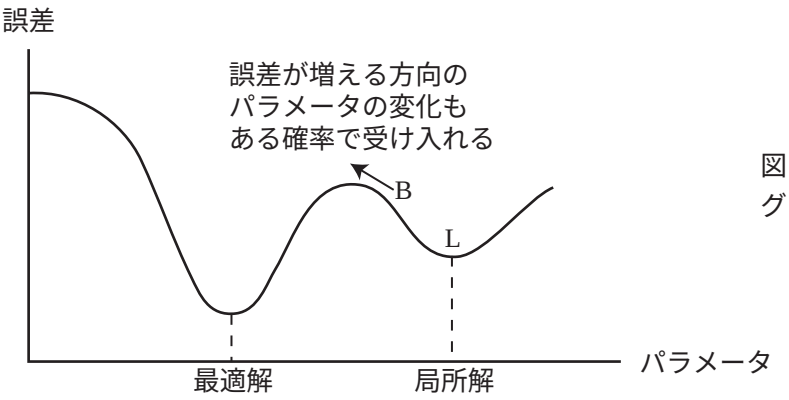

図２．シミュレーティッド・アニーリングによる局所解からの脱出

１．初期パラメータと初期温度 $T(0)$を定める．

２．現在のパラメータでの系の出力を求め，理想出力との誤差を求める．

３．パラメータを一部変更する．
例えば，結合重みを少し増やす／減らす，フィルタのウィンドウ形状を１画素変更する，などの操作を行います．

４．変更されたパラメータでの系の出力を求め，理想出力との誤差を求める．

５．変更前後で誤差を比較する．
(1) 変更後の方が誤差が減少している場合，その変更を受け入れる．
(2) 変更後の方が誤差が増加している場合，ある確率 p ($p \le 0.5$)その変更を受け入れる．受け入れない場合はその変更を取り消す．
確率 p は，例えば，現在の繰り返し回数を n，温度を $T(n)$，誤差の増加量を ΔE とするとき，シグモイド関数を使って

$$p = \frac{1}{1 + \exp(\Delta E / T(n))} \tag{1}$$

と定めます．温度 $T(n)$は，繰り返しにしたがって，例えば

$$T(n) = 0.99T(n-1) \tag{2}$$

のように低下させます．したがって，温度 $T(n)$が０に近づくほど，また誤差の増加量 ΔE が大きくなるほど，受け入れ確率は０に近づきます．

６．３．に戻る．

この過程を，それ以上パラメータの変更が受け入れられなくなるまで，あるいは十分な回数の繰り返しが進むまで繰り返します．

遺伝的アルゴリズム

シミュレーティッド・アニーリングは，１回のくり返しで１組のパラメータを試し，それによる誤差を見て解の探索を進めるものでした．そこで，１回のくりかえしで多数の組のパラメータを同時に試せば，もっと効率良く解の探索ができると考えられます．これを生物の進化に見たてて行う方法が，遺伝的アルゴリズム，あるいは進化論的アルゴリズムといわれるものです．

遺伝的アルゴリズムでは，個体(individual)と遺伝子(gene)でパラメータを表現します．遺伝子は 1/

0のビットの列で，これでひとつのパラメータを表します．個体はいくつかの遺伝子をもち，１つの個体が１組のパラメータに相当します．

さて，初期状態では，それぞればらばらの遺伝子をもつたくさんの個体が存在します．それぞれの個体で，遺伝子に対応するパラメータを用いて系の出力を求め，理想出力との誤差をそれぞれ求めます．そして，誤差の大小によって個体に優劣をつけ，劣った個体を淘汰します．淘汰によって残った優れた個体から，子孫をつくります．このときに，後述する遺伝子の交差や突然変異などによって，新しい遺伝子をもつ個体を作り出します．

この操作によって，もとの個体から次の世代の個体が生まれました．これを繰り返して世代交代を進めてゆくと，より小さな誤差を生じる「優れた」個体が徐々に増えてきます．しかも，世代交代の際，常に新しい遺伝子を持つ個体を作っているので，局所解に陥らず広い範囲でパラメータの探索を行うことができます．

遺伝的アルゴリズムの手続きは概ね以下のようになります．

１．パラメータをビット列による遺伝子で表し，遺伝子の組である個体を定義する．
例えば，3x3画素の構造要素を遺伝子で表現すると，図３のようにビット列で表すことができます．

２．適応度を定義する．
適応度が高いことは，個体が「優れている」ことを表します．今考えている，系の出力を最適化するパラメータを求める問題では，その個体のもつパラメータによる系の出力と理想出力との誤差が小さいとき，適応度が高くなるようにします．

３．遺伝子をランダムに生成し，ある数の個体を生成する．

４．各個体について適応度を求める．

５．淘汰 (selection) を行う．適応度の低い一定割合の個体を捨てる．

６．残った個体から，次の世代の個体を以下のような方法で生成する．
交差(crossover)－図４のように，親世代の２つの個体の遺伝子を混ぜ合わせて子世代の個体の遺伝子をつくります．遺伝子の各ビットでどちらの親の遺伝子を受け継ぐかは，あらかじめ決めておく，ランダムに選ぶ，などの方法で決めます．この操作で，淘汰によって残った個体の性質を受け継ぎかつ新しい遺伝子をもつ個体をつくることができます．

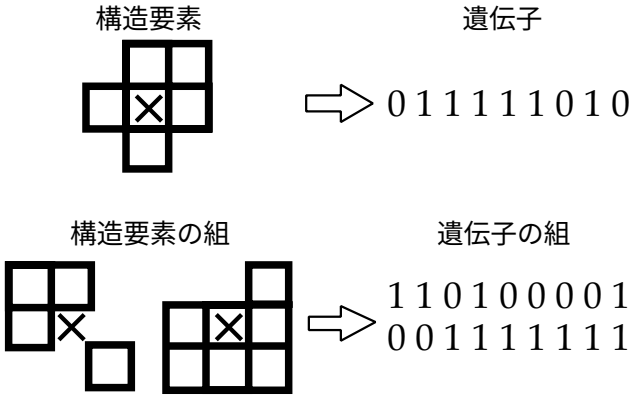

図３．遺伝子による表現（□：画素，×：原点）

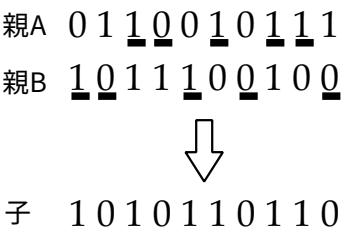

図４．交差

突然変異(mutation)－親世代の個体の遺伝子，あるいは交差によって生成された個体の遺伝子の各ビットを，ある小さい確率で反転します．この操作で，最適解の探索の範囲を広げ，局所解に陥らないようにすることができます．

なお，淘汰によって残った個体のうち適応度が上位のいくつかを，交差も突然変異もせずにそのまま次の世代に引き継ぐこともあります．これらは優れた個体なので，次の世代でこれらを上回る適応度の個体が現れるまでそのまま引き継ごうというわけです．

７．十分な適応度の個体が得られるまで，４～６を繰り返す．

この手続きをみてわかるように，遺伝的アルゴリズムは並列計算に適したアルゴリズムです．遺伝的アルゴリズム自体が考案されたのは結構昔のことですが，近年実用的な並列計算機が現れてきたことによって，注目を集めるようになっています．