2005 年 1 月 13 日

知能型システム論: 多層パーセプトロンの誤差逆伝搬学習

喜多 一

1 最適化問題としての学習

多層パーセプトロン (MLP) の学習とは

> 適当な入力データと望ましい出力データ（**教師信号**）を用意し，MLP にその入力データを提示したときの出力が教師信号になるべく一致するように結合重み w_{ij} を調整すること

を言い，この形式の学習は教師信号を与えることから**教師有り学習 (supervised learning)** と呼ばれる．

いま学習用に P 組の入力，教師信号データの対 $(\boldsymbol{x}_{\mathrm{I}}^p, \boldsymbol{d}_{\mathrm{O}}^p)$, $p = 1, \ldots, P$ を考える．ここで入力信号 $\boldsymbol{x}_{\mathrm{I}}^p$ は入力ユニット数 I に等しい次元のベクトル，教師信号は出力ユニット数 O に等しい次元のベクトルである．

p 番目のデータに対して MLP が生成する出力 $\boldsymbol{x}_{\mathrm{O}}^p$ の評価を教師信号との差の自乗を用いて

$$G_p = \frac{1}{2} \sum_{i \in U_{\mathrm{O}}} (d_i^p - x_i^p)^2 \tag{1}$$

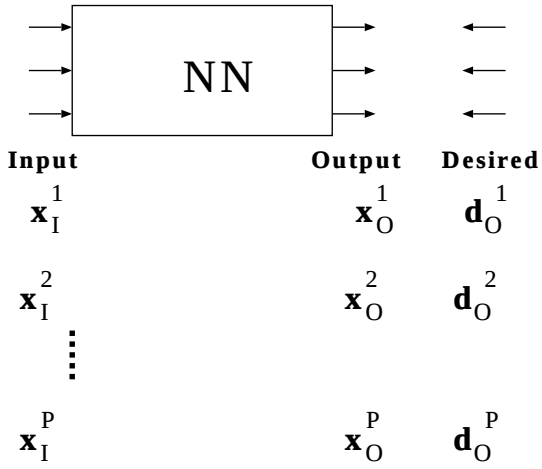

図 1　MLP の学習

添え字が多くて煩雑であるが，学習用のデータに関する添え字は肩につけて書く．

係数の 1/2 は 2 次関数を微分したときに係数をキャンセルするために導入されている．

とする．ここで U_{O} は出力ユニットの添字の集合，d_i^p, x_i^p はそれぞれ教師信号，出力層ユニットの第 i 成分である．これを P 個のデータすべてに対して加えたもの

$$G = \sum_{p=1}^{P} G_p = \frac{1}{2} \sum_{p=1}^{P} \sum_{i \in U_{\mathrm{O}}} (d_i^p - x_i^p)^2 \tag{2}$$

を学習用データ全体での MLP の評価値とし，これを結合重み $\boldsymbol{w}$ について最小化することを「学習」として定式化する．

すなわちネットワークの学習とは非線形最適化問題

$$\min_{\{w_{ij}\}} G = \min_{\{w_{ij}\}} \sum_{p=1}^{P} G_p = \min_{\{w_{ij}\}} \frac{1}{2} \sum_{p=1}^{P} \sum_{i \in U_{\mathrm{O}}} (d_i^p - x_i^p)^2 \tag{3}$$

を解くこととして定式化できる．

この考え方は観測データを説明する関数の決定に用いる最小自乗法と同じである．ただし，通常，最小自乗法で近似される関数，例えば 1 次近似 $y = w_1 x + w_0$ や 2 次近似 $y = w_2 x^2 + w_1 x + w_0$ は **パラメータ w_i について見るとその 1 次式になっており，この場合，最小自乗法は解析的に解を求めることができる．** これに対して，MLP では処理ユニットの非線形特性 (シグモイド関数) が介在するために数値的に最適化を行わなければならない．

2　勾配法による非線形最適化

パラメータ $\boldsymbol{w}$ に関して連続な非線形関数 $G(\boldsymbol{w})$ を $\boldsymbol{w}$ について最小化する基本的なアルゴリズムは $\boldsymbol{w}$ について何らかの初期値を与え，G が減少する方向に $\boldsymbol{w}$ を調整して行く方法である．G が微分可能な場合は G が最も減少する方向は勾配の逆方向 $-\nabla_{\boldsymbol{w}} G$ であることから

$$\boldsymbol{w}^{\mathrm{NEW}} = \boldsymbol{w}^{\mathrm{OLD}} - \alpha \nabla_{\boldsymbol{w}} G \tag{4}$$

として $\boldsymbol{w}$ を更新して行く方法が考えられ，勾配法とか最急降下法と呼ばれる．ここで α は更新の幅を表す正数である．

なお，勾配 $\nabla_{\boldsymbol{w}} G$ とはスカラーの関数 G をベクトル $\boldsymbol{w}$ の各成分で偏微分して得られるベクトル

$$\nabla_{\boldsymbol{w}} G = \begin{pmatrix} \frac{\partial G}{\partial w_1} \\ \frac{\partial G}{\partial w_2} \\ \vdots \\ \frac{\partial G}{\partial w_N} \end{pmatrix} \tag{5}$$

である．

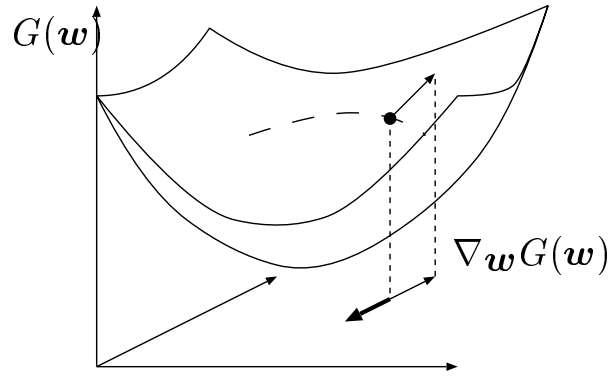

図 2　勾配法の考え方

NN ではベクトル $\boldsymbol{w}$ の添え字は 2 つあるが，ここでは表記のため 1 次元的に並べなおしておく．また勾配ベクトルは縦ベクトルとする．

3　演習

この演習は次節のための準備である.

1. 処理ユニットで用いられる非線形特性であるシグモイド関数

$$f(s) = \frac{1}{1+\exp(-s)} \tag{6}$$

について，その導関数 $f'(s)$ を求めよ．また求めた導関数は $f(s)$ を用いて表現できることを示せ.

2. 図 3 に示した多層パーセプトロンについて，単一パターンが与えられた場合の評価値 $G = \frac{1}{2}(d_4 - x_4)^2$ のパラメータ $\boldsymbol{w} = (w_{30}, w_{31}, w_{32}, w_{40}, w_{41})^{\mathrm{T}}$ に関する勾配

$$\nabla_{\boldsymbol{w}} G = \left(\frac{\partial G}{\partial w_{20}}, \frac{\partial G}{\partial w_{30}}, \frac{\partial G}{\partial w_{31}}, \frac{\partial G}{\partial w_{32}}, \frac{\partial G}{\partial w_{40}}, \frac{\partial G}{\partial w_{41}} \right)^{\mathrm{T}}$$

を求めよ．ただし処理ユニットの非線形特性 $f(s)$ に対して，その導関数は $f'(s)$ と表してよい.

導出の方針：まず，x_4 を $x_1, x_2, w_{30}, \cdots, w_{41}$ の関数として表し (中間ユニットの出力 x_3 を代入により消去する)，これを G に代入し，その後に $\boldsymbol{w}$ の各成分について偏微分を計算すればよい.

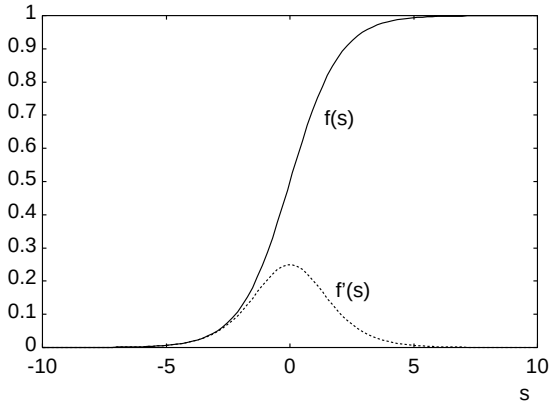

図 3　シグモイド関数

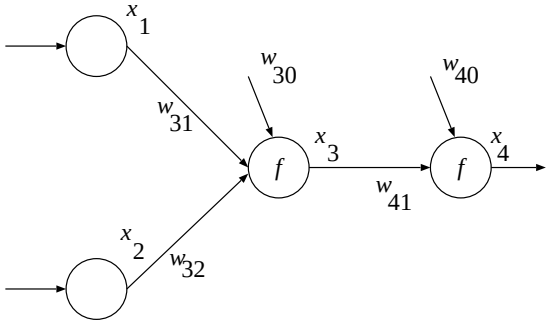

図 4　演習問題 MLP の構成

4　学習則の導出

本節では一般的な MLP の構成に対して勾配法による学習アルゴリズムを求める.

学習法は $-\nabla_{\boldsymbol{w}} G$ の方向に $\boldsymbol{w}$ を調整して行くことであるが $G = \sum_{p=1}^{P} G_p$ であるから，$-\nabla_{\boldsymbol{w}} G$ は，$-\nabla_{\boldsymbol{w}} G = -\nabla_{\boldsymbol{w}} \sum_p G_p = \sum_p (-\nabla_{\boldsymbol{w}} G_p)$ より $-\nabla_{\boldsymbol{w}} G_p$ を求めて総和すればよい．これは以下のように計算できる.

まず出力ユニットへの結合重みについては単純に

$$\delta_i^p \equiv -\frac{\partial G_p}{\partial s_i} = (d_i^p - x_i^p) {f_i}' \tag{7}$$

と定義すれば，ユニット i の動作は $x_i = f(s_i), s_i = \sum_j w_{ij} x_j$ であるから

$$-\frac{\partial G_p}{\partial w_{ij}} = -\frac{\partial G_p}{\partial s_i}\frac{\partial s_i}{\partial w_{ij}} = \delta_i^p x_j^p, \quad i \in U_{\mathrm{O}} \tag{8}$$

と求まる．ここで δ_i^p を**誤差信号**と呼ぶ.

隠れユニット (添字の集合を U_{H} とする) への結合重みについては MLP 上を信号が隠れ層から出力層へ伝搬することから微分のチェーンルールを用いて

$$-\frac{\partial G_p}{\partial w_{ij}} = -\left(\sum_{k \in U_{\mathrm{O}}} \frac{\partial G_p}{\partial s_k}\frac{\partial s_k}{\partial x_i}\right)\frac{\partial x_i}{\partial w_{ij}} = \left(\sum_{k \in U_{\mathrm{O}}} \delta_k^p w_{ki}\right) {f_i}' x_j^p = \delta_i^p x_j^p, \quad i \in U_{\mathrm{H}} \tag{9}$$

となる.

δ はデルタ（delta）とよむギリシャ文字．d を他の意味で使っているのでこれを使う.

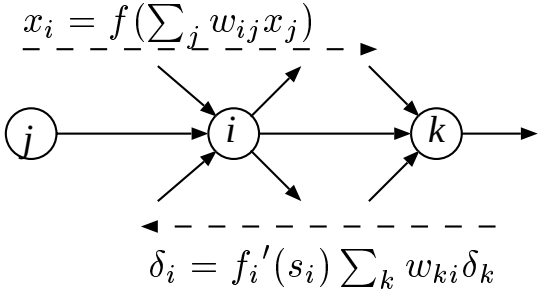

図 5　学習誤差信号の逆伝播

ここで誤差信号 δ_i^p は

$$\delta_i^p \equiv \left(\sum_{k \in U_{\mathrm{O}}} \delta_k^p w_{ki} \right) {f_i}' \tag{10}$$

と定義され，(8) 式, (9) 式 が同じ形式で記述できる．この計算は誤差信号 δ_i^p が MLP の出力側から入力側に逆向きに伝わるように計算されることから**誤差逆伝搬法 (Error Backpropagation 法，BP 法)** と呼ばれる．

勾配法の最も単純な実装は α を正定数 (学習係数) として w_{ij} を $-\partial G/\partial w_{ij}$ の α 倍だけ変更するもので，全データの提示後に一括して結合重みを調整する「バッチ型」の学習則を与える：

$$w_{ij}^{\mathrm{NEW}} = w_{ij}^{\mathrm{OLD}} - \alpha \frac{\partial G}{\partial w_{ij}} = w_{ij}^{\mathrm{OLD}} + \alpha \sum_{p=1}^{P} \delta_i^p x_j^p \tag{11}$$

さらに，$G = \sum_{p=1}^{P} G_p$ より，個々のデータの提示毎に結合強度を調整する「オンライン型」の近似的学習則が得られる：

$$w_{ij}^{\mathrm{NEW}} = w_{ij}^{\mathrm{OLD}} - \alpha \frac{\partial G_p}{\partial w_{ij}} = w_{ij}^{\mathrm{OLD}} + \alpha \delta_i^p x_j^p \tag{12}$$

なお，学習係数 α を固定する単純な実装の場合，α の値は試行錯誤による決定を要する．α が小さいと学習は安定的に行われるが収束が遅い．一方，α を大きくしすぎると $\boldsymbol{w}$ が振動的に変化し学習が収束しなくなる．

より層の多いネットワークではこの手続きを入力側に遡って適用すればよい．

「誤差逆伝搬法」は勾配の計算方法を指す場合と (11) 式, (12) 式, (14) 式 などの学習ルールを指す場合とがある．

結合強度 $\boldsymbol{w}$ の調整により以降の勾配が変化するため，近似であるが学習係数が小さければあまり問題とはならない

実際には α の値を調整してもこれらの学習法は収束がかなり遅い．学習を加速する経験的方法として，前回の修正方向も加味した次の方法が用いられ，**モーメント法**と呼ばれている：

$$w_{ij}^{\mathrm{NEW}} = w_{ij}^{\mathrm{OLD}} + \Delta w_{ij}^{\mathrm{NEW}} \tag{13}$$

$$\Delta w_{ij}^{\mathrm{NEW}} = -\alpha_1 \frac{\partial G_p}{\partial w_{ij}} + \alpha_2 \Delta w_{ij}^{\mathrm{OLD}} \tag{14}$$

ここで α_1, α_2 は正の定数であり，例えば経験的な推奨値として $\alpha_1 = 0.1$, $\alpha_2 = 0.9$ などが用いられている．

5　BP 法の実装・応用時の留意点

MLP を実装し，応用する際には留意点がいくつかある．ここではそれらを整理しておく．

5.1　MLP の応用方法について

- MLP は実際の現場で広範囲に応用が可能である．パターン認識などの本格的な応用だけではなく，簡単なデータ整形など小さな応用も有用であり，人手による試行錯誤よりも MLP の学習のほうが効率的なことも多い．
- MLP はその構成ユニットの特性から，データから隠れた構造を抽出する能力は高いが，学習したデータ点の影響が入出力特性にかなり大域的な影響を及ぼ

す．しかしながら，用途によっては学習用データの影響をその近傍に限定したい場合もある．このような用途には MLP よりも k-NN 法 [4] や RBF[2] など入力データと学習用データとの距離を基準に動作を決定する手法を用いるべきである．

- 多層の MLP を試す前に，学習の容易な隠れ層のないネットワークを試しておくことは有用である．
- パターン認識に直接，画像，音声などのデータを用いると MLP への負荷が高く，学習による機能獲得のために大量のデータも必要になる．適当な特徴抽出などを入力信号の前処理として行うことが現実的である．その際，画像の 2 値化など情報を大幅に失う前処理は MLP による機能獲得を困難にする恐れがあり注意を要する．
- 学習用データは，それが取り得るバリエーション (変形，汚れなどを含む) に注意し，可能な限り多く収集する．

5.2 MLP の構成について

- パターン識別など 0-1 の判断を出力に要求する応用では出力層にシグモイド関数を用い，関数近似など値域を限定しない応用では出力層には線形関数 $f(s) = s$ を用いる．
- シグモイド関数を (6) 式 の通りに実装すると，内部の指数関数の数値計算で

オーバーフローが生じやすい．シグモイド関数は分子分母に $\exp(s)$ を掛けて $\exp(s)/(1+\exp(s))$ とも書けるので，入力の符号に応じて定義を切替えればこれを回避できる．

- MLP の応用上の決定事項の 1 つに「隠れ層のユニット数」がある．これは汎化能力の問題と密接にかかわる．以下に紹介する汎化の問題とその方策を考慮してユニット数を決定するとよい．

5.3 MLP の学習の実施について

- 入力信号の変域は予め吟味し，特定の入力が極端な変域を取らないようにスケーリングなど行っておく．
- 結合重みの初期値は 0 の近傍，例えば $[-0.1, 0.1]$ の範囲でランダムに決める．結合重みが大きすぎるとユニットの飽和領域での動作を招き，$|f'(s)|$ が小さいため学習が困難になる．また，MLP は構造上，すべての隠れ層ユニットが対称である．結合重みのランダムな初期化は対称性を破るために必要である．
- MLP の誤差評価 G は非凸であり，一般に複数の局所最適解を持つ．勾配法を基礎とする最適化法では大域的最適解の発見は保証されない．結合重みの初期値をランダムに取り直して学習を繰り返すなどして対応する．
- 誤差評価 G の最小値は 0 であるとは限らない．MLP が教師信号を再現できない場合は正にとどまるが，その要因は 2 つある．1 つは MLP の自由度が不足

している場合，もう 1 つは教師信号に矛盾やノイズが含まれ，再現が不可能な場合である．どちらが生じているかの吟味が必要である．

- 学習の停止は，それ以上，学習を進めても効果がないことを条件にする．具体的には評価値 G が学習を進めても一定値以上減少しない，あるいは勾配 $\nabla_{\boldsymbol{w}} G$ のノルムが 0 に近いことなどを判定の基準とする．ただしオンライン型の学習やモーメント法などでは，これらの値が揺らぐので条件を試行錯誤的に決定をせざるを得ない．このほかに，一定回数の学習ごとにテスト用データによる評価を行い，テスト用データに対する評価の改善が見られなくなった時点で学習を終了させる方法もある．
- 前節で述べた学習法は小規模な応用ではそれなりに動作するが，学習係数や停止条件などの試行錯誤的調整を要する．大規模な MLP の応用では計算の効率化とパラメータ選定に要する手間の削減が求められる．これには以下に紹介するより洗練された最適化法の適用を検討すべきである．
- MLP の学習結果は学習に使用していないテストデータを用いて評価しなければならない．そのために予めデータを学習用とテスト用に分割して学習を実施すべきである．

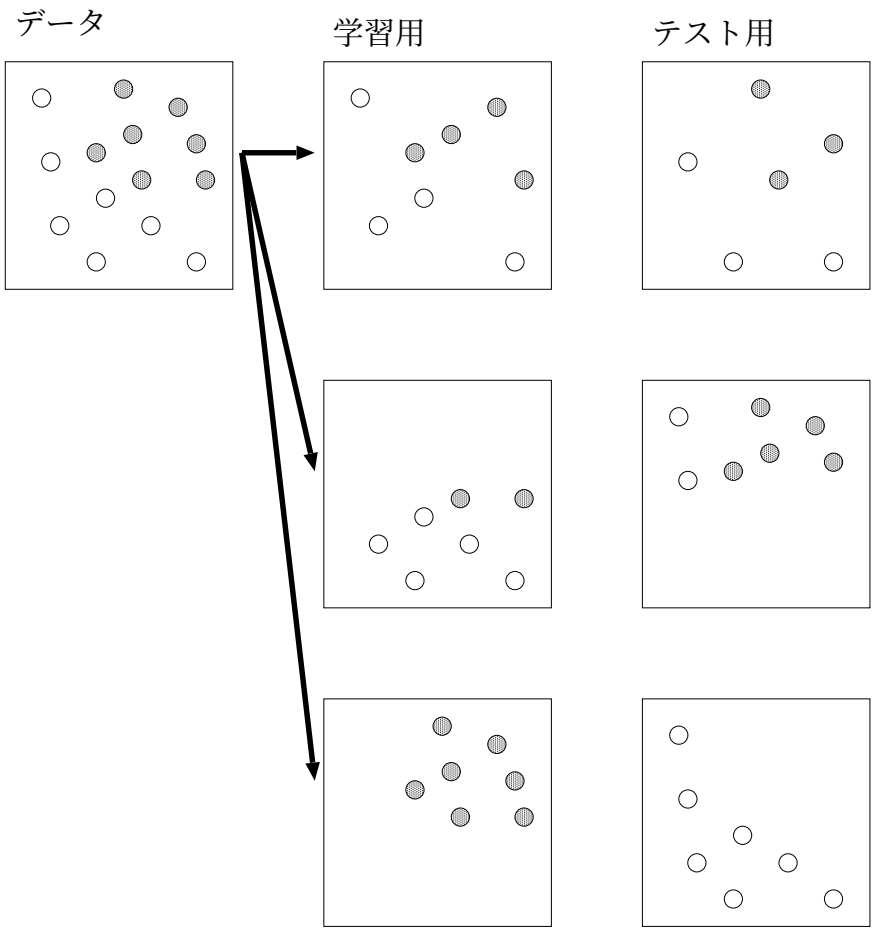

図 6　データの分割：よい分割と悪い分割

6　より洗練された最適化法の適用

2. 節で紹介した学習法は最適化法としては不十分なものである．その結果，1) 収束が遅く学習に時間がかかる，2) 学習係数などのパラメータを試行錯誤的に調整しなければならない，などの問題が生じる．実際の MLP の応用においては，得られた結果の吟味や隠れユニット数の調整などの作業に応じて学習を繰り返すので，これらの問題は研究・開発の効率をかなり低下させる．

そこでより洗練された非線形最適化手法を MLP の学習に適用ことがする考えられる [5] が，その際，以下の MLP 固有の事情を考慮する必要がある：

- 多くの非線形最適化手法は次のルールを繰り返し適用する：
  1. 現在の点からの探索方向を決定するルール,
  2. 探索方向に進むステップ幅を決定するルール

  後者には所定の探索方向での最適化を行う**直線探索** [3] がしばしば用いられる．非線形最適化手法の適用は厳密に勾配を求めるバッチ型の学習となるため学習用データ数が多いと 1 回の結合重みの評価にかなりの計算を要する．そこで直線探索をできるだけ少ない評価回数で済ませる工夫が必要である．
- MLP の結合重みの総数は容易に数百に達する．この場合，ニュートン法で用いられる目的関数の 2 階微分 (ヘッセ行列) など決定変数の 2 乗のオーダーの記憶量を要求する手法を単純に適用することは難しい．

これらを考慮した上で適用可能なアルゴリズムとして，**共役勾配法**がある [6]．これは 2 次関数に対して「共役方向」と呼ばれるベクトル群を用いて探索方向を決定し直線探索を行えば，高々変数の次元数だけの探索方向の設定で最適点に達する性質を用いた手法である．

また，高速で安定性も高い最適化法として**準ニュートン法**がある．しかし，この方法は変数の次元数の 2 乗のオーダの記憶量を要求するため，MLP には使いにくい．Saito らは準ニュートン法の BFGS 公式の生成を記憶量の制限を考慮して打ち切る手法を提案しており，良好な結果を得ている [7].

7　MLP の学習と汎化

7.1　学習と汎化

学習による機能獲得では

> 「与えられた学習用のデータを再現できた学習機械が未学習のデータに対しても妥当な応答を示せるか」

という**汎化**の問題は重要である．

汎化が議論できるためには

1. 学習課題には何らかの隠れた構造があり，そもそも汎化が期待できる必要がある．例えば氏名と電話番号のように，氏名から電話番号を推測する規則の存在

が期待できない学習課題では汎化の議論は無意味である.

2. 一方，学習する機械の側にも少ない自由度でデータを再現するという構造が必要である．単純に学習事例をメモリに蓄える形の学習では学習データの再現は容易であるが，未学習のデータに対する妥当な応答は期待できない.

7.2 MLP における汎化

MLP ではネットワークの構造に縛られて結合重みによってデータが学習される．これを通じてデータが持つ隠れた構造を獲得し，汎化を行おうというわけである．したがって

- 汎化のためには MLP の自由度 (隠れ層のユニット数や結合重み数) を拘束しつつデータの学習を行えばよい.
- 自由度が不足すると学習データを十分に再現できないが,
- 自由度が高すぎると学習データの再現が容易になりすぎ，未学習のデータへの応答の妥当性を保証できなくなる.

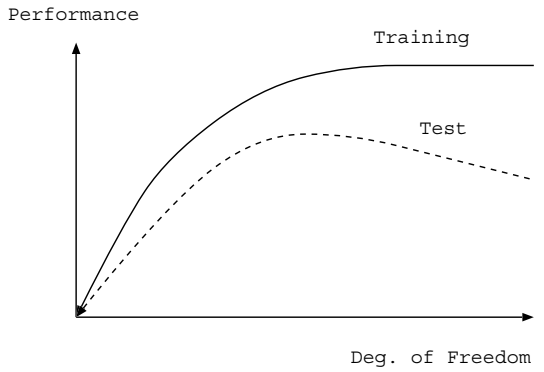

図 7　MLP の自由度と汎化能力

7.3 MLP のテスト

汎化のための基本的な技法に **MLP のテスト**がある.

- まず学習に際して，データを学習用とテスト用に分ける.

- そして，学習用データで結合重みを学習し，テスト用データで学習後の MLP を評価する．
- 隠れユニット数など MLP の自由度を高めて行くと学習データへの MLP の適合度は上昇する．一方，テスト用データに対する適合度は MLP の自由度が低い間は自由度の増加につれて向上するが，あるところからは適合度は向上しないか，むしろ悪くなる．
- そこでテスト用のデータに対する適合度の上昇が見られなくなったところを MLP の自由度として採用する．

学習用データの数が少ない場合には，より多くのデータを学習用に用いる Cross Validation 法 [2, 9] などが利用できる．その他，学習に際してデータへの適合とともに 0 でない結合重みをできるだけ少なくする手法など自由度の制御を中心として汎化のためのさまざまな技法があるが，これについては [9] を参照して頂きたい．

8 直接最適化法による学習

MLP の学習は教師信号の存在を仮定しているが，応用によっては明示的な教師信号が利用できないものもある．例えば MLP が埋め込まれたシステム全体の性能が与えられ，MLP の結合重みでこれを最適化したい場合である．このような場合には勾配も使いにくく直接法による最適化が適している．

最適制御や強化学習などの問題設定はこれに相当する．

8.1 順逆モデリング

MLP の後ろに未知システムが接続され，その出力に対する教師信号が与えられる場合を考えよう．このような状況は制御の問題で典型的に生じる．誤差逆伝搬法を用いるには未知システムの出力の入力に関する偏微分 (ヤコビ行列) が必要になる．しかしながら，一般に未知システムのヤコビ行列を求めることは難しいことが多い．この場合に MLP を用いる技法の一つとして順逆モデリング [10] がある．この手法ではまず，未知システムに MLP (順モデル) を並列し，未知システムの入出力関係を学習する．次に，学習した順モデルを用いて未知システムのヤコビ行列を近似的に求め前段の MLP (逆モデル) の学習を行おうというものである．

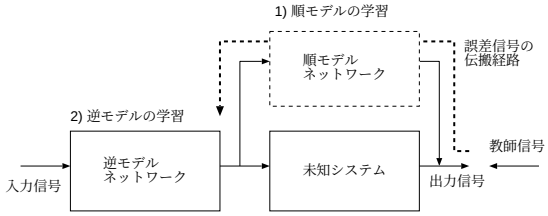

図 8　順逆モデリング

制御の問題では順モデルが未知システムの「入力 → 出力」を学習するのに対し，逆モデルは「望ましい出力 → それを生成するための入力」を表すことからこのように呼ばれる．

8.2 Powell の共役方向法による学習

MLP の結合重みの学習は最適化問題を解くことであると考えると必ずしも勾配など微分情報は必須ではない．勾配などが使いにくい状況では目的関数の評価値のみを利用する最適化法である直接法の利用が考えられる．直接法により効果的に最適化する手法として **Powell の共役方向法** [3] がある．これは 2 次関数についての共役方向を逐次生成し最適化を進める点では，先に紹介した共役勾配法と共通であるが，勾配情報を用いないという特色がある．中西らはニューラルネットワークによる制御系の学習に Powell 法を適用し良好な結果を得ている [11].

8.3 遺伝的アルゴリズムによる学習

近年，急速に進展している最適化法として**遺伝的アルゴリズム (Genetic Algorithms, GA)** がある．この手法では直接法による大域的な探索が可能であり，問題の構造に対する仮定が少なく，多様な評価関数の最適化に適用できる．ただし，標準的な GA は個体表現に 2 値遺伝子を用いるなど，連続変数の最適化には適さない．小野らは浮動小数点数を遺伝子表現とする実数値 GA に対して正規分布交叉 (UNDX) やその発展版を提案し，高い探索性能を有することを示している [12]．これらの交叉は多峰性関数の大域的探索だけでなく，変数間に依存関係のある非分離な関数の最適化も得意とする．MLP の学習にはこのような洗練された GA の構成法を用いる必要がある．

さらなる学習のために

解説書を中心に参考文献を挙げておく．MLP を含めニューラルネットワーク全般については [1, 2, 4] などを参照して頂きたい．また，非線形最適化の諸技法については [3] を参照頂きたい．

参考文献

[1] 西川, 北村編著：ニューラルネットワークと計測制御, システム制御情報ライブラリー 11, 朝倉書店 (1995).

[2] S. Haykin: Neural Networks, A Comprehensive Foundation, Macmillan (1994).

[3] 今野, 山下：非線形計画法, 日科技連 (1978).

[4] 上坂, 尾関：パターン認識と学習のアルゴリズム, 文一総合出版 (1990).

[5] R. Battiti: First- and second-order methods for learning between steepest descent and Newton's method, Neural Computation, Vol. 4, pp. 141-166 (1992).

[6] M. F. Moller: A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning, Neural Networks, Vol. 6, pp. 525-533 (1993).

[7] K. Saito and R. Nakano: Partial BFGS Update and Efficient Step-Length Calculation for Three-Layer Neural Networks, Neural Computation, Vol. 9, pp. 123-141 (1997).

[8] 黒江：リカレントニューラルネットワークの学習法, システム/制御/情報, Vol. 36, No. 10, pp. 634-643 (1992).

[9] 喜多：ニューラルネットワークの汎化能力, システム/制御/情報, Vol. 36, No. 10, pp. 625-633 (1992).

[10] 川人：脳の計算理論, 産業図書 (1996).

[11] 中西, 幸田, 井上：ニューラルネットワークによる最適フィードバック制御系の構成法, 計測自動制御学会論文集, Vo. 33, pp. 882-889 (1997)

[12] 小野, 山村, 喜多：実数値 GA とその応用, 人工知能学会誌, Vol. 15, No.2, pp. 259-266 (2000).