# 階層型ニューラルネットワークモデルの初期値の導出

前川眞一
(大学入試センター・研究開発部)

## 1　はじめに

９０年代初頭より、統計学においても、非線形の回帰分析や判別分析の手法としてニューラルネットワークモデルが利用されるようになってきている。ニューラルネットワークモデル、特に、一つの隠れ層を持つ３層パーセプトロンと呼ばれる前進型ネットワークモデルで第３層が線形のものは、$\boldsymbol{X}$ を説明変数行列 $(n \times q+1)$、$\boldsymbol{Y}$ を基準変数行列 $(n \times p)$、 $\boldsymbol{W}$ と $\boldsymbol{\theta}$ をモデルのパラメタとして、

$$\boldsymbol{Y} = F(\boldsymbol{X}|\boldsymbol{W},\boldsymbol{\theta}) + \boldsymbol{E} \tag{1}$$

という回帰モデルのモデル部分 $F$ に $\boldsymbol{X}$ のある特定の非線形関数

$$F(\boldsymbol{X}|\boldsymbol{W}) = \mathbf{1}\boldsymbol{\theta}^{3'} + \boldsymbol{\Psi}(\mathbf{1}\boldsymbol{\theta}^{2'} + \boldsymbol{X}\boldsymbol{W}^2)\boldsymbol{W}^3 \tag{2}$$

を用いたものであると言うことが出来る。ただし、

$$\boldsymbol{\Psi}(\boldsymbol{Z}) = \{\psi(z_{ij})\} \tag{3}$$

は要素ごとにいわゆるロジスティク変換を施す関数である。また、パラメタ $\boldsymbol{\theta} = \{\boldsymbol{\theta}^2, \boldsymbol{\theta}^3\}$, $\boldsymbol{W} = \{\boldsymbol{W}^2, \boldsymbol{W}^3\}$ のサイズは $n_1 = q$, $n_2, n_3 = p$ を各層のニューロン数とすれば、$\boldsymbol{\theta}^2, \boldsymbol{\theta}^3$ は $n_2 \times 1, n_3 \times 1$, $\boldsymbol{W}^2, \boldsymbol{W}^3$ はそれぞれ $(n_1+1) \times n_2, (n_2+1) \times n_3$ であり、その値は、回帰分析の場合には残差の２乗和を最少とするように、判別分析の場合には多項分布の尤度関数を最大とするようにし求められる。統計的手法としてニューラルネットワークをみた場合、それがどれだけ現有の手法に勝っているのか（もしくは劣っているのか）ということが興味の中心になるが、その比較を行うためには、ニューラルネットワークを手軽に使える環境が必要となる。しかし、ネットワークモデルの場合、そのモデルが非線形であるため、パラメタの推定に際しては何らかの数値解法を利用せざるをえず、また、その数値解法が洗練されていないため、解を得るために長時間を必要とする。たとえば、Mayekawa(1995) は、ネットワークモデルと伝統的統計的手法の比較のために、ネットワークモデルのパラメタの推定方法の改善策を示しているが、それを用いることにより単純な最急降下法を用いる場合と比べて 5 - 10 倍の収束速度の改善を得ている。しかし、それでもまだ遅いという印象は免れない。

本研究の目的は、ネットワークモデルの繰り返し計算を用いたパラメタの推定法の加速のために、このパラメタ $\boldsymbol{\theta}$ と $\boldsymbol{W}$ の良い初期値をいかにして簡単に求めるか、ということである。一般に、パラメタの推定において繰り返し計算を含む数値解法を利用する統計モデルは多数あるが、たとえば、因子分析の最尤解ではいわゆる主因子解を、多次元尺度法の最少２乗解では古典的多次元尺度法の解を用いるなど、その大部分は合理的な初期値 (rational initial) から繰り返し計算を始める。また、それらの初期値は解析的に（繰り返し計算なし

に）求めることが出来るものである。従って、乱数を用いた場合と比較して、格段に少ない時間で解（極値）に到達することが可能となる。

ネットワークモデルの初期値の問題 (network initialization) に関する研究はいくつか行われているが、それらは主に、いかにして「良い」乱数を生成するかに集中し、モデルに適した合理的な初期値を解析的に求めることを試みた研究は少ない。

本研究においては、３層モデルの場合、第１層 $\boldsymbol{X}$ から第２層への仮のパラメタ $\boldsymbol{\theta}^{2*}, \boldsymbol{W}^{2*}$ を、第２層の仮の出力

$$\boldsymbol{O}^{2*} = \boldsymbol{\Psi}(\mathbf{1}\boldsymbol{\theta}^{2*'} + \boldsymbol{X}\boldsymbol{W}^{2*}) \tag{4}$$

が広範囲の単調増加関数群を生成するようにシステマティックに作成し、第２層から第３層への重み $\boldsymbol{W}^3$ を、変数選択により選ばれた $\boldsymbol{O}^{2*}$ のサブセット $\boldsymbol{O}^2$ への重み

$$\boldsymbol{Y} \approx \mathbf{1}\boldsymbol{\theta}^{3'} + \boldsymbol{O}^2\boldsymbol{W}^3 \tag{5}$$

として決定するという方式を用いる。具体的には、$q$ 個の説明変数行列 $\boldsymbol{X}$ の各列およびそれらを直交化したものごとにおよそ $6 \sim 15$ の単調増加関数群を生成し、それらから変数選択を行う。従って、変数選択の問題は、$6$〜$15 \times 2 \times q$ の変数から効率よく第２層のニューロンの数 $n_2$ だけの変数を選択する問題となるが、ネットワークモデルの場合 $\boldsymbol{Y}$ は多変量である場合が多いため、残差積和行列のトレースを最少とするような説明変数群を選択していくことになる。

# 2　方法

## 2.1　ロジスティク基底の作成

与えられた説明変数 $\boldsymbol{X}$ から第２層の仮の出力 $\boldsymbol{O}^{2*}$ を作るためには様々な方法が考えられるが、ここでは以下のようにする。

1. 線形結合の作成
   $n_1 = q > 1$ の場合、$\boldsymbol{X}$ の各列の線形結合として、

   $$\boldsymbol{S} = [\boldsymbol{X} \quad \boldsymbol{F}] \tag{6}$$

   ただし、$\boldsymbol{F}$ は、列方向に中心化されたデータ行列の列平均を元のデータの右端に付け加えたもの

   $$[\boldsymbol{X} \quad \boldsymbol{J}\boldsymbol{X}\mathbf{1}] \tag{7}$$

   の $q$ 個の主成分スコア、という $n \times (2q = (Q))$ の行列を定義する。ここで列平均を加えるのは、あらかじめ $\boldsymbol{X}$ が直交する場合に $\boldsymbol{F} = \boldsymbol{X}$ となるのを防ぐためである。
   この行列の列平均と列標準偏差を $m_j, s_j, j = 1, 2, \ldots, Q$ とする。
2. ロジスティク展開
   $\boldsymbol{U}$ 行列の各列ごとに、あらかじめ決められた定数 $a_i, i = 1, 2, \ldots, n_a$ と $b_j, j = 1, 2, \ldots, n_b$ を用いて

   $$\boldsymbol{v}_{ijk} = \boldsymbol{\psi}\left(1.7\, a_i \left((\boldsymbol{u}_k - m_k\mathbf{1})/s_k - b_j\mathbf{1}\right)\right) \tag{8}$$

を横に並べた $n \times (Q \times n_a \times n_b)$ の行列を $\boldsymbol{O}^{2*}$ と定義する。なお、係数 $a_i, b_j$ としては、たとえば、

$$a = \{3.0, 2.0, 1.0\}, \quad b = \{-3, -2, -1, 0, 1, 2, 3\} \tag{9}$$

等を用いれば、十分に豊かなロジスティク基底を作成することが出来る。

この手続きにより $\boldsymbol{O}^{2*}$ 行列が得られたならば、

$$\boldsymbol{\Psi}^{-1}(\boldsymbol{O}^{2*}) = [\mathbf{1} \quad \boldsymbol{X}] \begin{pmatrix} \boldsymbol{\theta}^{2*'} \\ \boldsymbol{W}^{2*} \end{pmatrix} \tag{10}$$

を解けば仮のパラメタ $\boldsymbol{\theta}^{2*}, \boldsymbol{W}^{2*}$ が得られる。ただし、$\boldsymbol{\Psi}^{-1}(V)$ は $\boldsymbol{V}$ の各要素にロジスティク変換の逆を作用させる関数である。

### 2.2 変数選択

第２層の仮の出力 $\boldsymbol{O}^{2*}$ が与えられれば、第３層の重み $\boldsymbol{W}^3$ を決定するためには、 $\boldsymbol{Y}$ を $\boldsymbol{O}^{2*}$ に回帰すればよい。ただし、第３層のニューロンの数は $n_3$ と決まっているため、

$$\boldsymbol{Y} = \mathbf{1}\boldsymbol{\theta}^{3'} + \boldsymbol{O}^{2*}\boldsymbol{W}^{3*} + \boldsymbol{E} \tag{11}$$

というモデルから、もっともフィットの良い $n_3$ 個の変数のサブセットを選択しなくてはならない。そのための方法としては、もっとも単純なものは all-possible-regression であるが、今回の問題では説明変数行列 $\boldsymbol{O}^{2*}$ の列数が $(Q \times n_a \times n_b)$ と比較的大きいため、以下に示す前進選択の MAXR 法 (SAS Institute, 1982, p102) を多変量に拡張したものが適当であると考える。これは、厳密に残差行列のトレースを最小とするサイズが $n_3$ のサブセットを選択してくるものではないが、計算量が比較的少なく通常の前進選択法や後進選択法等よりもいい結果を与えることが多い。また、これを後進選択に変更して使うことも考えられる。

この手続きにより $\boldsymbol{O}^2$ 行列とそれに付随するパラメタ $\boldsymbol{\theta}^3, \boldsymbol{W}^3$ が得られるが、第２層へのパラメタ $\boldsymbol{\theta}^2, \boldsymbol{W}^2$ は先ほど得られた仮のパラメタ $\boldsymbol{\theta}^{2*}, \boldsymbol{W}^{2*}$ のサブセットとなっている。

### 2.3 ３層以上への拡張

層の数が３層以上の場合には、上部の中間層をのぞいてあたかも３層であるかのようにして上記の方法を用いればよい。たとえば４層の場合には、まず、第１層、第２層、第４層を用いて係数 $\boldsymbol{W}^2, \boldsymbol{W}^4$ を決め、その後、第２層、第３層、第４層を用いて$\boldsymbol{W}^3$ を決めることになる。

## 3 方法の評価

以下の関数 $(n_1 = q = 2, n_2 = 8, n_3 = 1)$ を使って方法の評価を行った。
harmonic function G3

$$y = 42.659(0.1 + x_1(.05 + x_1^4 - 10x_1^2x_2^2 + 5x_2^4)) \tag{12}$$

本文中に示した方法で求めたロジスティク基底行列 $\boldsymbol{O}^{2*}$ のサイズは$n_a = 3, n_b = 7$ で $441 \times 4 \times 3 \times 7 = (57)$ となる。これに定数項を加えた 58 の変数で求めた場合、rmse=0.60435, 変数選択の後に rmse=0.61667 を得た。

前進選択による MAXR 法

```
nextin   is the function which enter the best variable in OUT given IN.
replacer is the function which swaps the one in IN and the one in OUT.
m        is the total number of variables to be selected.

    free IN;    * list of variables currently in the model;
    OUT=1:nx;   * list of variables currently not in the model;

    do i=1 to m;

      * find the one which best reduces rss;
      run nextin( bestid1, Rss, C, OUT, locy, eps, 0<>(print-2) );
      print i "Variable" bestid1 "entered." Rss;
      if bestid1 ^= 0 then do;
       IN=IN||bestid1;
       * for each variable already entered,
         find a candidate not entered which replaces it;
       * This is order dependent.;

       do j=1 to ncol(IN);

         INj=IN[j];
         run replaceR( bestid2, bestRss, C, INj, OUT, locy, eps, 0<>(print-3) );
         * print i j "replace" bestid2 bestrss;
         if bestid2 ^= 0 & bestid2 ^= INj then do;
          * swap INj and bestid2:  take INj out, put bestid2 in;
          if ncol(IN) > 0 then IN[loc(IN=INj)]=bestid2;
          OUT[loc(OUT=bestid2)]=Inj;
          C=sweep(C, Inj||bestid2);
          Rss=sum(vecdiag(C[locy,locy]));
          if print >= 3 then
           print "Variable" INj "replaced by variable" bestid2 Rss;
         end;
       end;
       if print >= 2 then do;
        print "Best Model Found:" i Rss;
        print IN;
        Vn=shape(varnames[IN],1);
        print Vn;
       end;
       rssall=rssall//Rss;
       * sort IN;
       rank=rank(IN);
       dum=IN; dum[rank]=IN‘;
       INall[i,1:ncol(dum)]=dum;
      end;
      else goto done;

    end; * of m loop;

    done: *;
```