人　工　知　能　103－5
（1996．1．17）

ニューラルネットワークからの命題の抽出

月本 洋, 森田 千絵
(株) 東芝 研究開発センター

著者は以前ニューラルネットワークから命題を抽出する方法を提示した。その概要は以下のとおりである。定義域が離散的な場合にはニューラルネットワークが学習する関数は多重線形関数になる。その多重線形関数の空間はユークリッド空間になる。定義域が [0,1] の場合には上記のことが近似的に成立する。この多重線形関数を（連続）ブール関数で近似することによって命題を得る。そこではシグモイド関数を線形近似していたが、その後線形近似をしないアルゴリズムを考案したので、それについて述べ、Shavlik の方法と比較する。実験は *voting − records* と *iris* を用いて中間素子数、重みの初期値等の影響について述べる。

Extracting Propositions from Neural Networks

Hiroshi Tsukimoto, Chie Morita
Research & Development Center, Toshiba Corporation

Authors presented an algorithm for finding propositions from neural networks. The outline is as follows. When the domain is discrete, the functions that neural networks can learn are multi-linear functions. The space of multi-linear functions can be made into a Euclidean space. When the domain is [0,1], the above facts approximately hold. So, multi-linear functions representing units of neural networks can be approximated by (continuous) Boolean functions, where sigmoid functions are approximated by linear functions. We developed an algorithm without the linearlization. This paper describes the algorithm and also compares it with Shavlik's algorithm. Experimental results of *voting − records* and *iris* are presented.

1 まえがき

著者は以前ニューラルネットワークから命題を抽出する方法を提示した [7]。その概要は以下のとおりである。定義域が離散的な場合にはニューラルネットワークが学習する関数は多重線形関数になる。その多重線形関数の空間はユークリッド空間になる [7]。定義域が [0,1] の場合には上記のことが近似的に成立する。この多重線形関数を（連続）ブール関数で近似することによって命題を得る。そこではシグモイド関数を線形近似していたが [7]、その後線形近似をしないアルゴリズムを考案したので [8]、それについて述べ, Shavlik の方法と比較する。実験は *voting − records* と *iris* を用いて中間素子数、重みの初期値等の影響について述べる。

2 節では多重線形関数空間について述べる。3 節ではニューラルネットワークを（連続）ブール関数で近似する方法について述べる。4 節では Shavlik の方法を説明し、5 節で実験結果について述べる。

2 ブール代数の拡張としての多重線形関数空間

定義 1 n 変数多重線形関数 $f(x_1,...,x_n)$ の定義は以下の通りである。

$f(x_1,...,x_n) = \sum p_i x_1^{e_1} \cdot \cdot x_n^{e_n}$

但し p_i は実数、 x_i は変数、 e_i は 1 か 0 である。

例 2 変数の多重線形関数 $f(x,y)$ は $f(x,y) = pxy + qx + ry + s$ である。

定理 2 変数の定義域を $\{0,1\}$ とする。多重線形関数空間はブール関数のブール代数の原子で張られる線形空間である *[7]*。

定理 3 定義域を $\{0,1\}$ とする。次に内積を以下のように定義する。

$< f,g >= \sum_{\{0,1\}^n} fg$

多重線形関数空間はブール関数のブール代数の原子で張られるユークリッド空間になる *[7]*。

定理 4 定義域が $[0,1]$ の場合、内積 $< f,g >$ を次の様に定義すると多重線形関数空間はユークリッド空間になる。証明は *[3]* を参照。

$< f,g >= 2^n \int_0^1 \tau(fg)dx$

ただし $\tau(x^n) = x$ である。

ブール関数の定義域を $\{0,1\}$ から $[0,1]$ へ拡張した関数は古典論理の全公理を満たす [5] ので、本論文では連続ブール関数と呼ぶが、簡単のため「連続」を省く。以降、定義域は特に断らない限り、$\{0,1\}$ か [0,1] である。今までの議論で多重線形関数空間はユークリッド空間になったのだから、多重線形関数はベクトルで表現される。このベクトル表現を論理ベクトルと言う。以後、論理ベクトルは $\mathbf{f},\mathbf{g}$ とも (f_i), (g_i) とも書く。

3 ニューラルネットワークのブール関数近似

定理 5 ニューラルネットワークが学習できる関数は定義域が離散の場合は多重線形関数である。

証明 定義域が離散の場合はダミー変数を導入することで定義域を 2 値にできる。従って関数は n 変数の時は $\{0,1\}^n \rightarrow R$ となる。定義域が $\{0,1\}$ なので、$x^k = x(k \geq 2)$ が成立する。従って $\{0,1\}^n \rightarrow R$ の関数は多重線形関数になる。□

定義域が連続の場合は適当な方法で定義域を [0,1] にすることができる。定義域が [0,1] の場合には近似的に上述の定理が成立する。即ち

$$x^n = \begin{cases} x(n \leq a) \\ 0(n > a) \end{cases}$$

とする。但し a はある自然数である。

3.1 近似の基本方法

定理 6 多重線形関数を最も近いブール関数で近似することを考える。多重線形関数の論理ベクトルを (f_i) とし、ブール関数の論理ベクトルを $(g_i)(g_i = 0, 1)$ とする。近似方法は以下の通りである [4]。

$$g_i = \begin{cases} 1(f_i \geq 0.5) \\ 0(f_i < 0.5) \end{cases}$$

この近似法は無差別原理を用いると準最尤法であることが証明できる [6]。

3.2 多項式オーダーの近似アルゴリズム

前項の近似法の計算量は指数オーダであるので、ニューラルネットワークをブール関数で近似する多項式オーダーのアルゴリズムを開発した。この多項式オーダーのアルゴリズムはブール関数の形を DNF 式とし、低次から項を生成してゆき、ある次数で項の生成を打ち切る方法である。以前のアルゴリズムはシグモイド関数 $S(x)$ を線形近似していたが [7]、本アルゴリズムは線形近似していないので、より正確な命題を求められる [8]。

定理 7 ニューラルネットワークの素子を
$S(p_1x_1 + ... + p_nx_n + p_{n+1})$
とした時、近似後のブール関数に
$x_{i_1} \cdots x_{i_k}\overline{x}_{i_{k+1}} \cdots \overline{x}_{i_l}$
が存在する条件は以下の通りである。
$S(\sum_{i_1}^{i_k} p_{i_j} + p_{n+1} + \sum_{1 \leq j \leq n, j \neq i_1, .., i_l, p_j \leq 0} p_j) \geq 0.5$
但し $S(\cdot)$ は出力関数であるが、シグモイド関数である必要はない。出力関数に関する制約は広義単調であることである。即ち広義単調増加でも広義単調減少でも良く、不連続点があっても良い。

上記の判定条件を用いて低次の項から生成して行き、その項を論理和で接続して最終的なブール関数を得る。低次の項で存在が確認された変数に関してはその項より高次の項での存在の判定の必要がない。簡単な例で言えば x が存在することが分かれば xy, xz 等は存在する（$x = x \vee xy \vee xz$）ので xy, xz の存在の判定は必要ないのである。従ってこのアルゴリズムは簡単化を同時に行なっている。

4 Shavlik の方法について

Shavlik の方法の概要は ニューラルネットワークの学習を、通常良く使われるバックプロパゲーション法ではなく、Soft Weight Sharing と言う特殊な方法で行ない、学習後のニューラルネットワークから N of M アルゴリズムでルールを抽出するものである [1], [2]。

4.1 Soft Weight Sharing

この Soft Weight Sharing （以下 SWS) の基本的考え方はニューラルネットワークの重みの分布がいくつかの正規分布になるようにすることである。このようにすることによって N of M アルゴリズムでルールが抽出できるようになる。SWS の評価量は、バックプロパゲーション法などで使われる誤差だけでなく、複雑さの項を考慮にいれる。即ち評価量は誤差と複雑さの和である。これを最小にするような重み係数を探すが、これを共役傾斜法で行なう。なお細かいところはいくつかの版が存在し、現在のところ改良中であるとのことである。

4.2 N of M アルゴリズム

N of M アルゴリズムは以下の処理から構成される。

1. クラスタリング：ニューラルネットワークの素子の重みを、予め決められた値より近い重みがなくなるようにクラスタリングする。
2. 平均化：クラスタ内の重みの平均を取る。

3. 除去：親ノードを活性化させるのに必要でないクラスタを除去する。手法は以下の二つがある。

   (a) 手法 1: 親ノードのバイアスを超過するような活性の供給ができないクラスタを除去する。

   (b) 手法 2: 訓練パターンに影響を与えないようなクラスタを除去する。

4. 最適化: 重みを変更せずにバイアスだけをバックプロパゲーション法で学習しなおす。

5. ルール抽出：単純に子ノードの重みを足したものが、親ノードのバイアスを越える組み合わせを抽出する。

6. 簡略化： 以下のような N of M フォーマットに書換える。

   $X : -2$ of $\{a_1 a_2 a_3\}$

4.3 両手法の比較

N of M アルゴリズムの 5、6 項の部分が本論文のアルゴリズムの特殊な形であることは容易にわかる。従って、前節の手法は N of M アルゴリズムと類似性があり、この点に関しては N of M アルゴリズムの一般化になっている。Shavlik の方法はニューラルネットワークを SWS という特殊な方法で学習するため、バックプロパゲーション法等で学習されたニューラルネットワークからルールを抽出することはできない。また N of M アルゴリズムの最適化の処理で重みを変更せずにバイアスだけをバックプロパゲーション法で学習しなおすため、学習後のニューラルネットワークが与えられた時にそのニューラルネットワークからルールを抽出することはできない。即ち、以下の短所がある。

1. 学習方法が制限されている。

2. ニューラルネットワークの構成を変更せずにルールを抽出できない。

これに対し、本論文の方法は学習方法は任意であり、ニューラルネットワークの構成を変更せずに命題を抽出できる。特にニューラルネットワークの構成を変更せずに命題を抽出できるため、中間素子等がどのような命題を学習したかがわかるという長所がある。

5 実験

以下の実験での命題の項の生成は、1 次の項が生成されても、2 次の項を生成し、2 次の項が生成されたら停止する。2 次までで項が生成されない時は 3 次まで項を生成している。3 次までででも項が生成されない時は 0 とみなしている。理由は 1 次で項の生成を止めると正解率が悪いと予想されるし、4 次以上の項を生成しても複雑で理解できないと考えられるからである。

5.1 *voting-records*

これは 1984 年の米国議会議員の記録である。属性はいくつかの政策であり、属性値は政策に関する議員の賛否 (y,n) であり、以下の通りである。

1; handicapped-infants: y,n
2; water-project-cost-sharing: y,n
3; adoption-of-the-budget-resolution: y,n
4; physician-fee-freeze: y,n
5; el-salvador-aid: y,n
6; religious-groups-in-schools: y,n
7; anti-satellite-test-ban: y,n
8; aid-to-nicaraguan-contras: y,n
9; mx-missile: y,n
10; immigration: y,n
11; synfuels-corporation-cutback: y,n
12; education-spending: y,n
13; superfund-right-to-sue: y,n
14; crime: y,n
15; duty-free-exports: y,n
16; export-administration-act-south-africa: y,n

クラスは議員の所属する政党で民主党と共和党のどちらかである。例えば事例は次のようになる。

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	クラス
y	y	y	n	y	y	y	n	y	y	y	n	y	y	y	n	民主党

入力は 16 個であり、出力は 1 個である。学習方法はバックプロパゲーションで反復は 2 乗誤差 0.01 で停止している。学習には 232 個の事例を用い、予測も同じデータで行なっている。学習後のニューラル

ネットワークの予測正解率は100% である。中間素子数を2、3、4個、重み係数の初期値を3個で実験を行なった。その結果、得られた命題の正解率は表1の通りである。

表 1: 正解率

中間素子数	初期値 1	初期値 2	初期値 3
2	0.767	0.763	0.772
3	0.940	0.931	0.931
4	0.931	0.944	0.918

中間素子2個の場合はあまり良い結果が得られていないが、これは今後分析したい。中間素子3個の場合には民主党に関して以下の命題が得られた。共和党はその否定である。

初期値1 : $\bar{4} \cdot (3 \vee 13)$

初期値2 : $\bar{4} \cdot (10 \cdot \bar{11} \vee 3 \cdot \bar{11} \vee 3 \cdot 10)$

初期値3 : $3 \cdot \bar{4}$

中間素子4個の場合も同様な結果が得られたが省略する。中間素子3個の場合の学習結果を見ると、いずれの場合も3個の中間素子のうち2個の学習結果は0であり、無駄であることが分かった。そこで2個の中間素子を削除し、1個にして実験したが、中間素子1個では、次数3次までの命題は生成されなかった。中間素子1個は基本的には中間素子0個と同じであるので中間素子を0個として学習を行なった。その結果, 重みの初期値にかかわらず、得られた命題の正解率は0.931であった。そこで初期値1で学習の途中で得られる命題を調べた。その結果を表2に示す。

表 2: 途中経過

反復回数	正解率 (NN)	正解率 (命題)	命題 (民主党)
100	0.987	0.935	$\bar{4} \cdot 11$
500	0.996	0.931	$\bar{4} \cdot (3 \cdot \bar{11} \vee 3 \cdot 10 \vee 10 \cdot \bar{11})$
1000	0.996	0.931	$\bar{4} \cdot (3 \cdot \bar{11} \vee 3 \cdot 10 \vee 10 \cdot \bar{11})$
5000	0.996	0.931	$\bar{4} \cdot (3 \cdot \bar{11} \vee 3 \cdot 10 \vee \bar{9} \cdot \bar{11} \vee \bar{9} \cdot 10 \vee 10 \cdot \bar{11})$
10000	1.000	0.931	$\bar{4} \cdot (3 \cdot \bar{11} \vee 3 \cdot 10 \vee \bar{9} \cdot \bar{11} \vee \bar{9} \cdot 10 \vee 10 \cdot \bar{11})$

上記の例ではニューラルネットワークの正解率は反復回数10000回で1になるが、命題は反復回数1000回以降は同じである。そして中間素子3個の場合とは違う命題で、若干複雑な命題が得られる。この例では中間素子の3個のうち2個は0を学習したが、その2個を外して学習を行なうと違う結果が得られた。もちろん学習後は0を学習した中間素子は不要なので削除することはできる。また反復回数100回の命題が正解率が1番良くかつ1番簡単であると言うことも興味深い。今後、どのようにすれば簡単で正確な命題が得られるかを考えたい。

5.2 *iris*

iris のデータは、がく片の長さ, がく片の幅、花弁の長さ、花弁の幅という4つの連続属性で事例を表し、クラスは iris(あやめ) の種類で、*setosa*、*versicolour*、*virginica* の3個である。たとえば、事例は以下のように表されている。

がく片の長さ (1)	がく片の幅 (2)	花弁の長さ (3)	花弁の幅 (4)	クラス
7.0	3.2	4.7	1.4	versicolour

入力は4個であり、出力は3個である。学習方法はバックプロパゲーションで反復は2乗誤差0.01で停止している。学習には150個の事例を用い、予測も同じデータで行なっている。なお定理5での a の値は十分多くとっている。すなわち基本的に線形近似している。学習後のニューラルネットワークの予測正解率は100% である。中間素子数を0、2、3個、重み係数の初期値を2個で実験を行なった。その結果、得られた命題の正解率は表3の通りである。

表 3: 正解率

中間素子数	初期値 1	初期値 2
0	0.800	0.800
2	0.973	0.973
3	0.973	0.667
4	0.667	0.973

上記表より分かる通り、中間素子数 0 の時の正解率は良くない。また中間素子が 3 個の初期値 2 と中間素子 4 個の初期値 1 の場合は正解率が悪いがこれは *versicolour* に関して 3 次までで命題が生成されていないので 0 にしているためである。従って 4 次の命題を生成すれば正解率が良くなると考えられる。得られた命題は例えば中間素子数 3 で初期値 1 の時は以下の通りである。

中間素子 1(t_1) : $3 \cdot 4 \vee \bar{2} \cdot 4 \vee \bar{2} \cdot 3$
中間素子 2(t_2) : $\bar{3} \vee 2 \cdot \bar{4}$
中間素子 3(t_3) : $\bar{2} \cdot (\bar{1} \vee 3)$
出力素子 1 (*setosa*) : $\bar{t_1} \cdot t_2$
出力素子 2 (*versicolour*) : $t_2 \cdot t_3$
出力素子 3 (*virginica*) : $\bar{t_2} \cdot \bar{t_3}$

出力素子に中間素子の結果を代入すると以下のようになる。

setosa: $\bar{3} \cdot (2 \vee \bar{4})$
versicolour : $\bar{1} \cdot \bar{2} \cdot \bar{3}$
virginica: $2 \cdot 3 \cdot 4$

6 おわりに

定義域が離散的な場合にはニューラルネットワークが学習する関数は多重線形関数になり、その多重線形関数の空間はユークリッド空間になる。定義域が [0,1] の場合には上記のことが近似的に成立する。この多重線形関数を (連続) ブール関数で近似することによって命題が得られる。この考えに基づいたアルゴリズムを提示し、Shavlik の方法と比較した。実験は *voting − records* と *iris* を用いて中間素子数、重みの初期値等の影響について述べた。

参考文献

[1] M.W. Craven and J.W. Shavlik: Learning symbolic rules using artificial neural networks, *Proceedings of the Tenth International Machine learning Conference*, pp.73-80,1993.

[2] J.W. Shavlik: Combining Symbolic and Neural Learning, *Machine Learning*, 14, pp.321-331, 1994.

[3] 月本 洋: 命題論理の幾何的モデル, 情報処理学会論文誌, Vol.31, pp.783-791, 1990.

[4] 月本 洋: 確率データからの帰納学習, 人工知能学会誌, Vol.7, No.5, pp.870-876, 1992.

[5] 月本 洋: 古典論理の全ての公理を満たす連続値論理について, 電子情報通信学会論文誌, Vol.J77-D-I No.3, pp.247-252, 1994.

[6] 月本 洋, 下郡信宏, 森田千絵: 回帰分析に基づく帰納学習アルゴリズム, 情報処理学会研究報告, 94-AI-96, 96-2, 1994.

[7] 月本 洋: パターン処理の近似としての記号処理, 電子情報通信学会論文誌, Vol.J78-D-II No.2, p.333-339, 1995.

[8] 月本 洋: ニューラルネットワークの論理的分析, 人工知能学会研究会資料、SIG-FAI-9501, pp.25-32, 1995.