

自己組織化原理に基づく交通制御手法に関する研究

—ニューラルネットワークモデルの交通制御への適用—

Self-Organizing Approach to Traffic Control Problems
Using Neural Network Models

中 辻 隆*

by Takashi Nakatsuji

To realize a self-organizing traffic control, neural network models were introduced. First, a four-layered neural network model was applied to a problem of traffic responsive selection of pre-stored timing plans. The neural model forms the input-output relation between incoming traffic flows and an advisable timing plan. Not only results by simulation and optimization analyses, but also experiences of traffic operators are able to be built in the model. Next, a four-layered neural network model, consisting of both a training process and an optimization process, was applied to a problem of estimation of optimal splits. A stepwise method that combined the Cauchy machine with a feedback method was able to produce a good approximated sequence of splits for a road network system.

1. まえがき

今日、世界の多くの都市が慢性的な交通渋滞問題に直面している。交通マネジメントシステムは、現有施設を空間的、時間的に有効利用することによってこの問題に対処しようとするものである。将来の交通マネジメントシステム、特に交通制御システムにおいては、制御方法を交通状況に応じて自律的に変化させる「自己組織化制御」がその基本的理念となっている¹⁾。これまでも、ファジィ論理²⁾やエキスパートシステム³⁾を応用することによって自己組織化能力を有する交通制御システムの開発が試みられてきている。

ニューラルネットワークモデルは、エキスパートシステムにおけるif-then 演算のような論理的演算を必要とせずに、過去の入出力の結果のみから自律

的に学習を行うことが出来るので、非線形、非定常、あるいは非論理的構造の表現能力に優れていると言われている。ニューラルネットワークモデルのこのような特性を、交通制御に関わる諸問題、例えば、交通変量の短期予測、非計測変量の推定、最適制御、あるいはインシデントの検知等に応用することにより、効率的な施設の運用を計ることが出来る。

筆者らは現在、ニューラルネットワークモデルを組み込んだ巨視的交通流シミュレーションプログラムの開発を行っている。これまでに、計測交通変量の短期予測、単独交差点における最適スプリットの推定、あるいは交通配分問題への適用を行い^{4) 5)}、所要の成果を得てきた。ここでは主に

- 1) 制御パターンのプログラム選択
- 2) 最適スプリットの推定

の2つの問題に対して階層型のニューラルネットワークモデルの適用を試みる。前者においては、プログラムの選択基準を交通状況に応じて自律的に変えながら最適な制御パターンを選択する。そこでは、

* 正会員 工修 北大講師 工学部土木工学科
(〒060 札幌市北区北13条西08丁目)

管制官の経験等も容易に学習モデルに組み込むことが出来る。後者においてはまず、行列長最小規範によるスプリットの最適化を多交差点問題に拡張した。さらに、遅れや停止回数の最小といった他の制御規範に対してもニューラルネットワークモデルによる学習が可能であることを示した。ここでの特色は、状態変量の推定が容易であること、そして再学習によってスプリットの再調整機構も自律的に変化させることが出来ることなどである。

2. ニューラルネットワークモデル^{8) 7) 8)}

ここでは、階層型ニューラルネットワークモデルとその学習アルゴリズム、逆伝搬法について簡単に紹介する。

2.1 ニューロン

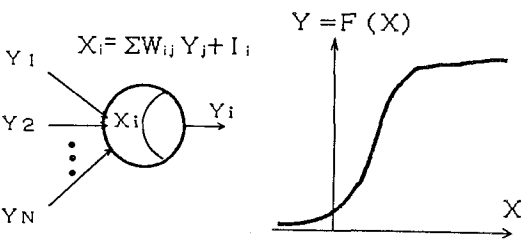
図-1 (a) に示すように、 Y_j を j 番目ニューロンから i 番目ニューロンへの入力、 I_i をバイアス入力、そして W_{ij} をその間の結合の強さを表すシナプス荷重とする時、時刻 n における i 番目ニューロンからの出力 $Y_i(n)$ は

$$X_i(n) = \sum_j W_{ij} Y_j + I_i(n) \quad (1)$$

$$Y_i(n) = f [X_i(n)] \quad (2)$$

と与えられる。ここで、関数 f は、応答関数と言われ、指数関数や正接関数などで表現される非線形単調増加関数 (図-1 (b)) である。ここでは以下のシグモイド関数を用いた。

$$f(x) = 1 / [1 + \exp(-x)] \quad (3)$$



(a) ニューロン (b) 応答関数
図-1 ニューラルネットワークモデル

2.2 階層型ニューラルネットワークモデル

本論文においては、図-2 に示すような、4 層から成る階層型のニューラルネットワークモデルを用いた。A 層を入力層、B と C の 2 層を中間層、そして D 層を出力層とした。A 層は、入力情報を B 層に伝える分配器としての役割を担う。B 層は、入力データを正規化したり、あるいは統合・集約を行い次の C 層にデータを伝える。そして D 層において目的変量を出力する。すなわち、A、B、および D 層のニューロンの数は問題に応じて自動的に決まる。C 層のニューロン数のみ自由に変えることが出来る。

多層型ニューラルネットワークモデルの学習アルゴリズムに逆伝搬法がある。逆伝搬法においては、出力層の結果を既知の教師信号と比較し、誤差の度合に応じて各層間のシナプス荷重を調整する。上に述べたように、A B 間の結合は自動的に規定されるので、シナプス結合の大きさが学習を通して調整されるのは B C 間と C D 間の結合のみである。逆伝播法による学習は、D 層における出力信号と教師信号の誤差の自乗和を最小化することによって行われる。B C 間、C D 間のシナプス荷重を W_{ij} 、 W_{jk} とする時、それらの調整量は

$$\delta W_{jk} = (z_k - y_k) y_k (1 - y_k) y_j \quad (4)$$

$$\delta W_{ij} = \sum_k (z_k - y_k) y_k (1 - y_k) W_{jk} y_i y_j (1 - y_j) \quad (5)$$

として求めることができる。実際の計算においては演算の収束速度と安定を高めるためにモーメント法と呼ばれる平滑化演算を行う。

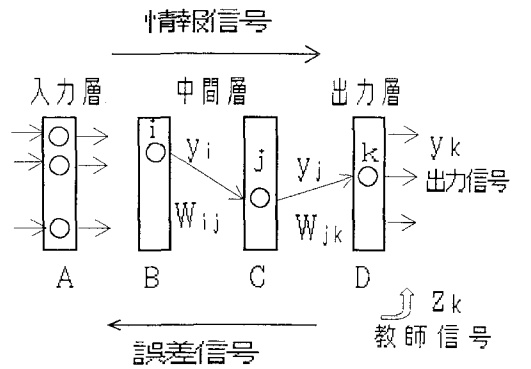


図-2 階層型ニューラルネットワークモデル

3. プログラム選択制御

ここでは、プログラム選択制御システムにニューラルネットワークモデルを導入する。単独交差点を例として説明する。

3.1 プログラム選択制御

通常のプログラム選択制御においては、感知器データの計測値をもとに選択レベルの算出を行い、サイクル長、スプリット、およびオフセットパターンの選択を個々に行う。ここでは、問題を単純化するために制御パラメータを組み合わせた制御パターンを準備し交通状況に応じて適当なパターンを選択するとした。さらに、現実の交通制御からは乖離していることを承知の上で現示の構成自体も時間的に変動するものとした。図-3に今回解析例として取り上げた単独交差点の形状を示す。5つの流入部を持つ変形交差点である。設計時間交通量データ、あるいは前日における観測結果等の事前データをもとに、可能な現示案の検討を行う。ここでは、最適制御シミュレーションプログラム TRANSYT/9⁹⁾を用いて遅れと停止回数の線形加重和を最小とするよう現示案と制御パラメータの設定を行った。図-4に示すような2組の現示案(A, B)と2組のサイクル長(80秒、90秒)を組み合わせた4組の制御パターン(A-80, A-90, B-80, B-90)を設定した。なお、現示案Bでは、わが国では用いられない右折先出しを想定しているが、あくまでも一つの解析例である。

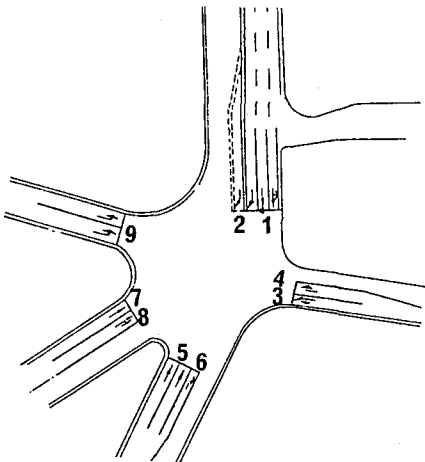


図-3 5枝交差点

3.2 制御パターン選択モデル

ニューラルモデルとして4層の階層モデルを用いた(図-2)。A層への入力はある流入リンクにおける予測流入交通量である。制御パターン選択の時間遅れを避けるためには、出来る限り正確に予測された交通量データを用いることが望ましい。ここでは、予測精度による影響を避けるために、正確な予測値が得られているものと仮定した。A層からB層への情報伝達をどのように集約統合するかによって多様な選択モデルを作成することが出来る。例えば、通常のパターン選択システムで用いられているM値や選択レベル値もB層の出力とすることができる。

ここでは、出力変量が制御パターンという質的データであることを考慮して、以下の2変量

- ・ 2つの主要流入リンクの正規化交通量の差
- ・ 各現示の飽和度の最大値

に対して予め閾値を設定し、A層からB層への入力値がその閾値より小さければ0、反対に大きければ1として2値化演算を行った。2つ閾値は、後で述べるようにニューラルモデルの再学習過程において交通状況に応じて再調整される。

D層からの出力は、目的とする制御パターンであり個々のニューロンがパターン案に対応する。D層に与える教師信号としては、現示案に対応するニューロンには1、他のニューロンには0を割り当てる。なお、これまでの経験に基づきC層はD層と同じニューロン数を持つとした。

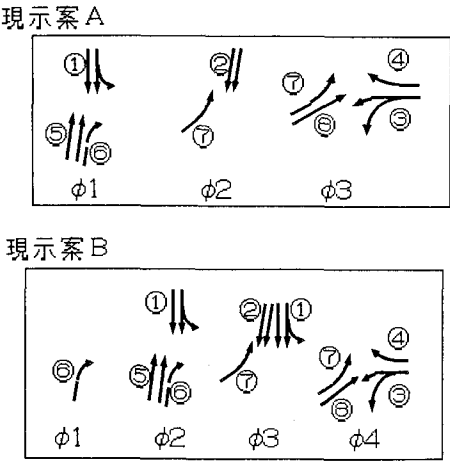


図-4 現示構成案

図-5 (a) に、リアルタイム方式による制御パターン選択システムの基本的処理フローを示す。機能的に3つの過程から構成される。

- 1) 初期学習過程：初めにシナプス荷重を乱数によって初期設定する。次に既知の交通量データと制御パターンの組み合わせを教師信号としてシナプス荷重の調整を行う。
- 2) パターン選択過程：A層に交通量データを入力し順次D層まで式(1)(3)による積和演算を行う。D層において最大値をとるニューロンを制御パターンとする。
- 3) 再学習過程：流入交通量の確定値を用いて、最適化シミュレーションを行ったり、あるいは管制官の判断に基づいて最適な制御パターンを決定する。それを教師信号としてシナプス荷重、あるいは閾値の再調整を行う。

初期学習においては、各学習データに対して、D層の出力値と教師信号との誤差自乗和が十分小さくなるまで逆伝搬法を繰り返す(図-5 (b))。B層における情報の統合において、M値等の連続値データを用いる場合には、各現示案に対応する1組以上の流入交通量を学習データとして準備する必要がある。それに対して本例のように2値化量を用いる場合には、B層から直接0、1の信号を入力して学習を行うことが出来るので、交通量データを事前に準備する必要がない。

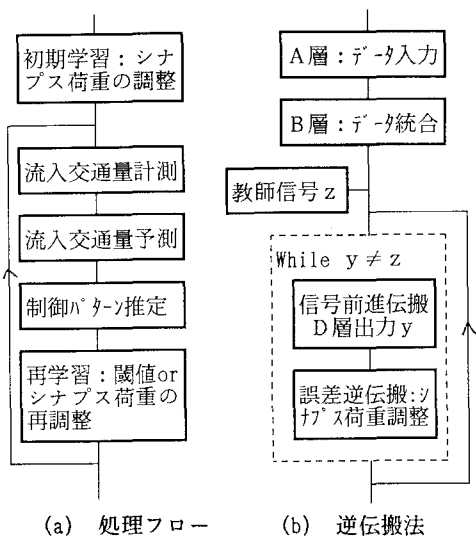


図-5 制御パターン選択モデル

図-6 に、逆伝搬法においてシナプス荷重が収束するまでの反復回数と学習次数の関係を示した。すなわち、各制御パターンを含む1組4つのデータを用いて15回(延べ60回)の学習を行った時に、各学習時に何度の収束演算を繰り返したかを示している。学習を通してシナプス荷重の大きさが直線的に改善されること、そして10数回の反復学習で十分満足のゆく収束解が得られることが示されている。なお、ここでは、D層における推定誤差の期待値が1%以下となることを収束判定基準とした。

選択された制御パターンが、シミュレーション解析や管制官の判断に基づく制御パターンと異なる時には再学習過程においてシナプス荷重や閾値の調整を行う。誤選択の原因がB層における2値化のパターンとD層における制御パターンとの対応の悪さにあると判断される場合には、新しい対応を設定しそれに基づいてシナプス荷重を再調整する。逆に、B層において2値化する時の閾値に問題があると判断される場合には、閾値を再調整する。いずれの方法で再調整するかは予めプログラム化しておく必要がある。ここでは、後者の方法、すなわち、閾値を修正する方法を採用している。実際の計算においては、閾値の急変を避けるために過去の値と移動平均した値を新しい閾値としている。

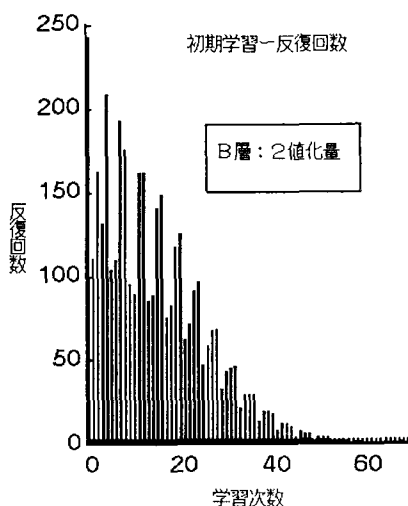


図-6 初期学習におけるシナプス荷重の収束状況

実際の利用においては、初期学習が十分に行われているならば、再学習をリアルタイムで行う必要はなく、時間帯、あるいは日単位ごとの再学習で十分である。逆に、再学習をリアルタイムで行うのであれば、最適化シミュレーションの結果に基づく制御を行えばよく、ニューラルネットワークモデルを導入する必要はない。また、本システムにおいては、教師信号としては、その交通状況に対して適切な制御パターンのみを指定するだけであるので、管制官の経験に基づく判断をシナプス荷重や閾値の調整に容易に取り込むことができる。

3.3 数値解析例

午前のピーク時間帯を対象として5分間交通量を順次与えて制御パターンをリアルタイムで推定した結果を表-1に示す。上段には各時間刻み毎に再学習を行った場合を、中段には、初期学習のみ行い再学習は行わなかった場合を、そして下段には最適化シミュレーションTRANSYT/9による結果を示した。現示案AとBが交互しているのが、実際の運用には問題があるが、交通状況が変動する8時台においても、比較的適切に制御パターンの推定を行っている。また、最初に仮定した閾値の設定が良かったこともあり、再学習なしでも良い推定を行っている。参考までに、推定された現示案の有効性を調べるために、微視的交通流モデル TRAF-NETSIM¹⁰⁾を用いてシミュレーション解析を行った。前・後半部をそれぞれ一つの現示案（前半A-80、後半B-80）に固定した場合に比べて、表-1の現示案（再学習あり）では、全遅れ時間が4.1%、車1台当たりの遅れ時間が7.4%改善することが確認された。

表-1 制御パターンの推定

時 刻	7:30-8:00	8:00-8:30	8:30-9:00	9:00-9:30	正 答 率
再 学 習 有	□○□□□□	●●○□●○	□□○■□■	□□□□□□	21/24
〃 無	□○□□□□	●●■□●■	□□○■□□	□□□□□□	21/24
TRANSYT9	□○□□□□	●●□□●○	□□○□□□	□□□□□□	

○ : A-80sec ● : A-90sec □ : B-80sec ■ : B-90sec

4 スプリット最適制御

いくつかの交差点から構成されるネットワークにおいて、各交差点のスプリットを最適化するニューラルネットワークモデルについて説明する。ただし、ここではオフセットの最適化は行わないものとする。

4.1 状態方程式

サイクル長単位に離散化された時間軸をもつ交通ネットワークにおいて、制御変量（スプリット）と状態変量（行列長、あるいは遅れ時間等）の時系列ベクトルを $u(k)$ 、 $y(k)$ 、 $(k=1, 2, \dots, N)$ とする。固定した時間区間 $0 < t < N \Delta t$ において制約条件

$$0 \leq y(k) \leq Y_{max} \tag{6}$$

$$U_{min} \leq u(k) \leq U_{max} \tag{7}$$

のもとで目的関数

$$J = \sum \| y(k) \|^2 \tag{8}$$

を最小とするような制御変量 $u(k)$ を求める問題を考える。この時、状態変量 $y(k)$ は、 $u(k)$ や流入交通量 $q(k)$ を説明変数とする状態方程式によって記述される。例えば、各流入リンクにおける行列長を状態変量とする時には、状態方程式は、

$$y(k+1) = y(k) + B_0 u(k) + B_1 u(k-1) + \dots + B_m u(k-m) + q(k) \tag{9}$$

と表現され、初期値を与えることによって行列長の時系列値を算定することができる。ここで、 B_j は制御重み行列と言われ、各流入部における飽和交通流率により定義される（定式化の詳細は文献(12)）。

各流入部における遅れや停止回数を状態変数とする時には、サイクル長をさらに時間的に細分化したステップ単位に状態方程式を作成する。時刻ステップ m における流入交通量 (IN Profile) と上流交差点からの流出交通量 (OUT Profile) を各々 $IN(m)$ と $OUT(m)$ とする。この時状態方程式が

$$IN(m+u) = F \cdot OUT(m) + (1-F) \cdot IN(m+u-1) \quad (10)$$

と表わされるので、1 サイクル毎のシミュレーションを行うことによってそのサイクル中の遅れ時間や停止回数を算定することが出来る。ここで、 u は上流リンクからの平均走行時間であり、 F は車群の拡散を表す係数である。なお、Flow Profile 挙動に基づく遅れや停止回数の算定法は文献(9)に詳しい。

4.2 最適スプリット推定モデル

最適スプリットの時系列値を推定するためのニューラルネットワークモデルを図-7に示す。これまでと同じく4層の階層構造とした。A層からは制御変数、スプリット $u(k)$ の時系列値を入力する。 r 番目交差点の現示数を p_r とするとA層は全部で $\sum p_r N$ 個のニューロンから構成される。B層においては情報の統合は行わずA層からのデータをそのままC層へ伝える。D層からは状態変量 $y(k)$ の時系列値を出力する。 r 番目交差点の流入リンク数が n_r であるときには、D層は全部で $\sum n_r N$ のニューロンを持つ (C層もD層と同じニューロン数)。

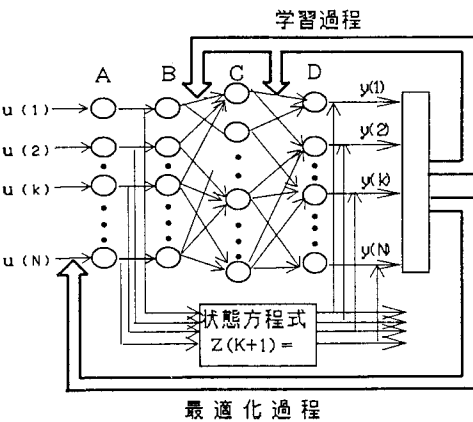


図-7 最適スプリット推定モデル

本システムは、2つの過程から構成される。一つが各交差点における交通流の動的挙動を表現するための学習過程であり、他方がスプリットを最適化するための最適化過程である。学習過程においてはまず、入力変量（スプリットの時系列値）を乱数によって初期設定する。次に逆伝播法を用いて、出力層からの出力変量 $y(k)$ が学習信号 $z(k)$ に一致するようにシナプス荷重の大きさを調整する。この時学習信号 $z(k)$ は状態方程式に基づいて算出される。スプリットの初期値を変えて学習を繰り返すことによって安定した入出力関係を形成することが出来る。

こうした学習過程を形成することにより単純な積和演算のみで状態変量を推定することができるようになる。巨視的交通流モデルにおいては、制御変量の微調整による状態変量の再計算に多くの演算時間を費やしている。本モデルを交通流モデルに組み込むことにより演算時間の短縮を計ることが出来る。

図-8は、最も単純な4枝交差点を対象として、1) 行列長を状態変量とした場合と、2) 遅れ時間を状態変量とした場合について、逆伝播法による推定誤差の変動を表したものである。ここでは、初期値をランダムに5回 (1, 3, 5回のみ表示) 変更して学習を行っているが、数度の学習でもかなり推定誤差が小さくなっている。実際の交通変動においてはスプリットがランダムに大きく変動することはないので、一度初期学習を行えば、かなりの期間再学習なしに状態変量の推定が出来ると思われる。

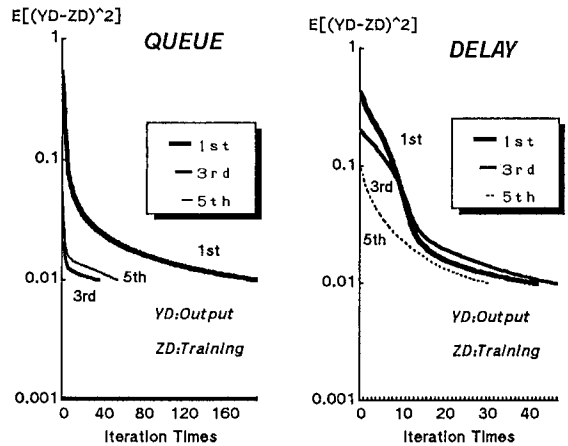


図-8 待ち行列（左）と遅れ（右）の学習

最適化過程においては、与えられた制限条件の下で目的関数を最小とするよう制御変量を調整する。この調整は、シナプス荷重の調整と同じように目的関数、式(8)を制御変量 $u_i(k)$ で偏微分することによって求めることが出来る。すなわち、

$$\delta u_i = \eta \sum_k y_k^2 (1 - y_k) \sum_j W_{ij} W_{jk} (1 - y_j) \quad (11)$$

ここで、 y_i 、 y_j 、および y_k は、図-2に示すように、それぞれB、C、D層における出力値である。また、 η は学習係数と呼ばれる定数である。ただし、この調整は、シナプス荷重の調整における逆伝搬法とは異なり直接A層にフィードバックされる。ここでは、この算法をフィードバック法と呼ぶ。このように制御変量の調整量をシナプス荷重を用いて表現することによって、シナプス荷重の変化に連動して、制御変量もその推定係数を変化させることが出来る。すなわち、自己組織化能力を持つことを表している。これが、ニューラルネットワークモデルによるスプリット最適化の特色である。

4.3 数値計算例

Singhら¹²⁾によって与えられた数値計算例を用いて解析解との比較を行った。図-9に交差点の形状、流入リンク、および各交差点における現示方式を示した。各リンクにおける飽和交通流率等の基本的入力値については文献(12)に詳しい。最適化は行列長最小規範に則って行った。実際の計算においては、ローカルミニマムへの収束を避けるとともに演算速度を早めるために、コースマシーン⁶⁾と上記フィードバック法を段階的に組み合わせたステップワイズ法⁵⁾を用いた。コースマシンの導入によりグローバルミニマムへの収束が促進され、スプリットの推定値が解析解によく一致していることが示されている。紙数の制限から図示できないが、行列長も解析解によく一致している。

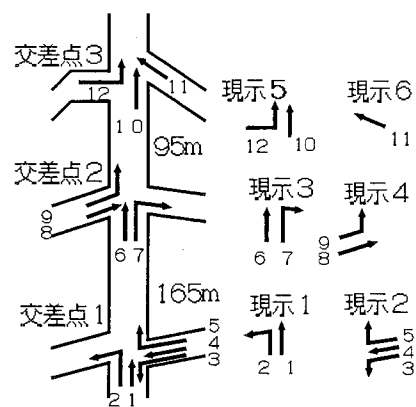


図-9 交差点形状と現示 (Singh et al.⁽¹²⁾)

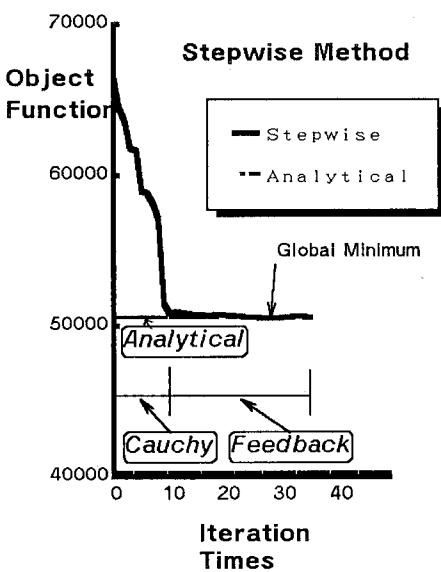


図-10 ステップワイズ法による最適解への収束

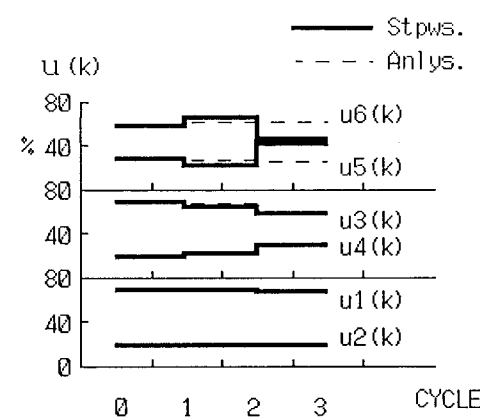


図-11 最適スプリットの推定結果

5. あとがき

ニューラルネットワークモデルの代表的モデルである階層型モデルをプログラム選択制御問題とスプリット最適制御問題に適用した。前者においては1) 入力変量を2値化してプログラム選択を行うモデルの提案を行い、初期学習が十分に行われていれば、再学習なしに比較的良好な推定結果を与えることを示した。本モデルにおいては、入力変量の2値化分類が適切に設定できれば、初期学習に事前の交通データを必要としないこと、再学習時に管制官の経験に基づく判断も教師信号として容易に取り込むことができることを特色としている。

後者においては、

2) 従来の単独交差点における行列長最小化規範に基づく最適化問題⁵⁾を多交差点の問題に拡張して数値解析を行い、解析解と一致することを確認した。
3) 本モデルが、行列長だけでなく、遅れや停止回数などといった他の状態変数に対しても十分な学習能力を有することを示した。

ここでは、交通状況の変動に応じてシナプス荷重が調整された時に、スプリットの調整係数も自動的に変化させることができる。

しかしながら、今後に残されている課題も多い。一番大きな問題はオフセットの最適化である。ここと同じ多層型のモデルを導入して交通状態と経験的に結び付けて推定するシステムも考えられるが、組み合わせが膨大となり学習することが困難と思われる。ホップフィールド型のモデルの適用を検討して行きたい。また、プログラム選択における交通状況の分類を0と1の2値量と表現するのではなく、ファジィ数として表現することもあわせて検討して行きたい。

ニューラルネットワークモデルを組み込んだ交通流シミュレーションプログラムは多くの優れた機能を有する。それは、簡単な積和演算による状態量の推定能力であり、ローカルミニマムの回避能力であり、演算速度の向上(並列処理能力)であるが、最も大きなものは過去の経験を取り込む学習能力である。こうした交通流シミュレーションプログラムの実現によって、交通管制官がリアルタイムにシミュ

レーション解析を行いながら制御パラメータを検討する。そしてその結果を学習機能を通してモデルにフィードバックさせる。そのようなシステムが近い将来作られるものと思われる。

最後に、本研究の実施に当たり、解析解の利用に便宜をお計り下さった大阪大学・田村坦之教授と有益な助言をして下さった北海道大学・加来照俊教授に記して謝意を表します。

参考文献

- 1) OECD: Dynamic Traffic Management in Urban and Suburban Road Systems, 1987
- 2) 秋山他: ファジィ流入制御モデルを用いた交通制御方法の評価と検討, 土木学会論文集, 413号/IV-12, 1990, pp. 77-86
- 3) R. Sigua et al.: An Expert System for Intersection Design and Control, 土木計画学研究・講演集, No. 12, 1989, PP. 275-282
- 4) 中辻他: ニューラルネットワークモデルの交通工学への適用について, 土木計画学研究・講演集, No. 12, 1989, PP. 297-304
- 5) T. Nakatsuji and et al: Application of Neural Network Model to Traffic Engineering Problems, 11th Int. Symp. Transp. Traffic Theory, 1990(in press)
- 6) P. D. Wasserman: Neural Computing, Van Nostrand, 1988
- 7) D.E. Rumelhart et al.: "Learning internal representations by error propagation, in Parallel Distributed Processing Vol. 1, MIT Press, 1986
- 8) 麻生: 「ニューラルネットワーク情報処理」、産業図書、1988
- 9) Crabtree: TRANSYT/9 User's Manual, TRRL, 1988
- 10) TRAF-NETSIM User's Manual, FHWA, 1989
- 11) 前田他: ヒトの上肢多関節運動を学習・生成する多層神経回路モデル, 信学技法, MBE87-133, 1988
- 12) M. G. Singh and et al: Modeling and hierarchical optimization for over-saturated urban road traffic networks, Int. Journ. Control, 20, No. 6, pp. 913-934, 1974