(32) ニューラルネットワークを用いた構造非線形の最適化解析

Nonlinear Structural Optimization Using Neural Network

萩原　一郎（東工大・工学部）　　施　勤忠（東工大・工学部）
Ichiro HAGIWARA,　　Qinzhong SHI

In the process of optimization, expensive finite-element method (FEM) software is often applied to solve engineering design problems. To obtain the optimal solution requires an extreme numbers of iterations between the analysis software and optimization program. The work is time-consuming due to the amount of computer time required. Therefore, instead of using the FEM analysis, an approximate method using sensitivity analysis has been proposed to solve this problem. However, it is difficult to obtain an explicit form of sensitivity for the nonlinear dynamic system. While the difference method is usually used for the sensitivity analysis, it is not the most efficient method to greatly reduce the calculation. In this paper, the authors propose an efficient approximation method to simulate the objective function using the holographic neural network (HNN). This network exhibits a large learning capability and dynamic features of memory, of a two-degree-of-freedom nonlinear dynamic system. The optimization approach is simplified and the tremendous calculation expense is decreased, because the objective function in design areas can be approximately calculated using the trained neural network.

Key Words : Nonlinear Optimization, Holographic Neural Network, Approximate Theory, Computational Mechanics, Structural Analysis

1. はじめに

ｴﾈﾙｷﾞｰ環境問題中心に，構造の軽量化への要求は厳しさの度を高め，自動車車両などの衝突要件を適切に満たす設計法を得ることはますます困難となってきている。このようなことから，衝突解析モデルは一層詳細なものとなる傾向がある[1]。そのため発達したソフトウエアやハードウエアの援用によっても，例えば前面衝突の１回の解析計算時間は10時間程度となる。これでは，設計仕様案の衝突特性の予測はできてもその最適化や適性化は困難である。したがって衝突問題への最適化解析技術の援用が望まれており既に１階感度を使用する近似モデル最適化法の適用例がある[2]。その際、非線形の衝突解析では感度係数の陽な表現は困難であるため，設計変更したときの諸量と変更前の諸量の差を取り差分形式で感度係数を求める手法が取られている。このように差分で感度を求めるのでは計算時間は膨大となる。そこでニューラルネットワーク（NN）の近似能力を利用して目的関数そのものを近似することによって最適値を得る方法が考えられる。例えば，前面衝突ではフロントエンドのｴﾈﾙｷﾞｰ吸収量が目的関数となる。そこで極力少ない設計変数の組み合わせを入力にしそれぞれのｴﾈﾙｷﾞｰを出力とする学習を行い，その他の組み合わせは予測する。すなわちｴﾈﾙｷﾞｰの関数形状そのものを近似するわけである。ここで注意したいのは本手法は，極大点及び極小点の数とおおよその位置があらかじめ分かっているときにのみ利用できるということである。現実にはフロントエンドで吸収されるｴﾈﾙｷﾞｰ関数はフロントエンドの常識的な変更範囲ではすなわち根本的な設計変更をしない限り，極値点の数とその位置を始めその形状はほぼ一定である。そして最適化解析を行う者がそれを知っているとする。本報ではこの条件下で所期の結果を得るためのNNを利用する最適化手順を提案する。なおNNとしては今日，バックプロパゲーション（逆伝播法）ニューラルネットワーク（BNN）が最もよく利用されているが，例えば，文献(3)の例題では学習を行う範囲は極めて狭いにも関わらず相当の学習回数を要し，予測範囲も極めて狭いものでしかない。これはBNNの能力不足とも考えられる。そこで，ここでは，神経診断[4]，パターン認識[5]などの問題に利用されて学習精度，速度ともにBNNより優れていると評価されているホログラフィックニューラルネットワーク（HNN）[6]の利用を図る。HNNがこれまで非線形最適化解析に利用されなかったのは，神経診断，パターン認識などといったファジイ分野に比べ，写像関数の選択や入力の拡張法がより困難であるためである。本報ではこれらの点に留意して，簡易モデルでフィージビリティスタディを行い，構造の非線形最適化解析の分野においてもHNNが極めて有効であることを示す。

2. ニューラルネットワークの関数近似理論

2．1 逆伝播法ニューラルネットワーク（BNN）　人工知能を模擬したNNは，非線形関数の近似計算，動的なシステム同定，パターン認識，振動騒音の制御，モデリングなどさまざまな分野に使われている。その中でもBNNが最もよく利用されている。それは，BNNが任意の関数に対して近似できる特徴を持つためである。中間層は１つで，$\mathbf{P}$ を入力ベクトル，R をベクトルの成分数とする。S_1 は中間層ニューロンの数，S_2 は出力層ニューロンの数，$\mathbf{W}_1$，$\mathbf{W}_2$ はシナプスの刺激を表す重み係数，$\mathbf{b}_1$ と $\mathbf{b}_2$ はそれぞれ中間層と出力層のバイアス値，括弧内はﾍﾞｸﾄﾙ或は行列の大きさを示す。BNNでは入力データ $\mathbf{P}$ に重み係数 $\mathbf{W}_1$ をかけて内積を $\mathbf{W}_1*\mathbf{P}$ で表わし，バイアス値 $\mathbf{b}_1$ を加えて $\mathbf{H}_1=\mathbf{W}_1*\mathbf{P}+\mathbf{b}_1$ をニューロンへ入力する。そして，重み $\mathbf{W}_2$ をかけバイアス値 $\mathbf{b}_2$ を

加えて出力層へ伝達する。最後に，出力層を経て出力する。BNNでは，NNからの出力 $\mathbf{H}_2$ と望ましい出力或いは教師データ $\mathbf{T}$ との差 $\mathbf{E}$（誤差関数）を最小にするように重み係数およびバイアス値を決めてゆく。通常，ニューロンの伝達関数として式(1)に示すようなシグモイド関数がよく使われている。

$$sigmoid(x) = \{1 + \exp(-x)\}^{-1} \in [0,1] \quad \cdots\cdots (1)$$

このようなシグモイド関数の有限なN個の組み合せで任意の関数を表せることは数学的に証明されている。すなわち，$f(x)$ を連続関数とすると，任意の小さい $\varepsilon>0$ に対し，実数N，$C_i, \theta_i \in R$，$w_i \in R^n$，$C_i = C_i[f(x)]$，i=1,...,Nが存在し，次の式が成り立つ。

$$\left| f(x) - \sum_{j=1}^{N} C_j sigmoid(y_j * x + \theta_j) \right| < \varepsilon \quad (x \in [0,1]^n) \quad \cdots (2)$$

ただし，$f(x) \in [0,1]$，$x \in [0,1]^n$（n次元の空間）である。ここで，y*xはyとxの内積である。このタイプの近似式理論は，階層型NNの数学理論となる。中間層を有する3層BNNは任意の非線形連続関数を中間層ニューロンの数を増やすことによって任意精度で近似できることが証明されている[7]。また文献(8)では中間層ニューロンの数を適切に選ぶ基準が検討されている．しかし，実用上では，以下の問題が存在すると考えられる。

1）重み係数およびバイアスの初期値は，通常，乱数で与えられるので，最適化計算の収束はこれらの初期値に依存する。すなわち，適切なスタート点でないと局所最小値に陥る可能性がある。

2）重み係数およびバイアスを決める学習は，陰関数の最適化過程を要し，大規模な問題では相当の演算時間を要する。

3）中間層のニューロン数は基底関数の種類に関係するものであり，各現象に対して適切な基底関数を決めることは困難であるため，通常，中間層のニューロン数Nを増やして誤差関数を減らして行く。そのため，近似関数が対象関数に比べて複雑になり，汎化能力が劣化することになる[9]。

4）求める重み係数の数とバイアスの数を足しあわせたものより多くのパターン数の学習を必要とする[8]。そのため，予め多くの解析データ数の準備を要する。

2. 2 ホログラフィックニューラルネットワーク(HNN)

HNNは，Sutherlandによって開発された[6]が，構造の非線形最適化分野への応用例は見られない。それは，第1章で述べた理由以外に一連の論文を見てもその理論背景が必ずしも十分に説明されていないこともその理由になっていると思われる。そこで，著者らで検討，考察した内容を少し詳細に述べてみる。 HNNは，入力および出力の実数データを複素数平面へ写像することにより，入力と出力の関係を線形で表せることを利用している。すなわち，図2に示すようにHNNの仕組みはBNNのとは異なる。同図で，$\mathbf{P}$ を入力ベクトル，Rをベクトル成分数とする。括弧内はベクトル或は行列の大きさを示す。そして $\mathbf{X}$ はニューロンの伝達関数である。まず，非線形の変換関数により入力と出力データは複素数平面へ変換される。

$$f(s_k^j) = \lambda_k^j e^{i\theta_k^j} \quad \cdots\cdots (3)$$

$$g(r_k^l) = \gamma_k^l e^{i\varphi_k^l} \quad \cdots\cdots (4)$$

ここで，fは入力の写像関数，gは出力の写像関数で，kはサンプル番号に対応するインデックス，jとlはそれぞれ入力変数，出力変数に対応するインデックスを表している。θ_k^j，φ_k^l は写像関数により，変換される位相角度で，その範囲は[0～2π）である。λ_k^j と γ_k^l はそれぞれ入力変数，出力変数の対応する位相角領域に出現する確率[0～1]を表す。伝達関数[X]はネットワークの出力[A]と望ましい出力，すなわち，教師データ[T]との差

$$Err = ([A]-[T])^H([A]-[T]) \quad \cdots\cdots (5)$$

が最小となるように決められる。ここで，記号Hは共役転置を意味し

$$[A]=[S][X] \quad \cdots\cdots (6)$$

である。式(5)を最小にする条件から，式(7)が得られる。

$$[X] = ([S]^H[S])^{-1}[S]^H[T] \quad \cdots\cdots (7)$$

ここで，伝達関数[X]はニューロンシナプスの繋がりを数学的に表すもので，ニューロンの基本メモリであり，[S]と[T]はそれぞれ入出力のﾏﾄﾘｸｽである。式(8)，(9)の中で，p，n及びmはそれぞれサンプル数，入力および出力の数を表す。入力数が多くなり式（7）のﾏﾄﾘｸｽ[S]が大次元となると，

同式の逆ﾏﾄﾘｸｽを作成する演算回数はGauss-Jordan消去法

$$[S] = \begin{bmatrix} \lambda_1^1 e^{i\theta_1^1} & \lambda_2^1 e^{i\theta_2^1} & \cdots & \lambda_n^1 e^{i\theta_n^1} \\ \lambda_1^2 e^{i\theta_1^2} & \lambda_2^2 e^{i\theta_2^2} & \cdots & \lambda_n^2 e^{i\theta_n^2} \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_1^p e^{i\theta_1^p} & \lambda_2^p e^{i\theta_2^p} & \cdots & \lambda_n^p e^{i\theta_n^p} \end{bmatrix} \quad \cdots\cdots (8)$$

$$[T] = \begin{bmatrix} \gamma_1^1 e^{i\varphi_1^1} & \gamma_2^1 e^{i\varphi_2^1} & \cdots & \gamma_n^1 e^{i\varphi_n^1} \\ \gamma_1^2 e^{i\varphi_1^2} & \gamma_2^2 e^{i\varphi_2^2} & \cdots & \gamma_n^2 e^{i\varphi_n^2} \\ \cdots & \cdots & \cdots & \cdots \\ \gamma_1^p e^{i\varphi_1^p} & \gamma_2^p e^{i\varphi_2^p} & \cdots & \gamma_n^p e^{i\varphi_n^p} \end{bmatrix} \quad \cdots\cdots (9)$$

で n^3 回必要となる。それを避けるために次の反復計算法が利用される。まず，初期値の計算は式(10)を利用する。

$$[X] = [S]^H[T] \quad \cdots\cdots (10)$$

式(10)に基づく基本学習の精度が，十分でないとき，メモリ $\mathbf{X}$ を更新する学習により能力向上が図られる。次に，この点について述べる。新たな入力（刺激）[S']を与えたときのニューロンの対応する出力（応答）[R']は式(11)で得られる。

$$[R'] = \frac{1}{E}[S'] \cdot [X] \quad \cdots\cdots (11)$$

ここで，

$$E = \sum_{i=1}^{n} \lambda_i^{2} \qquad \cdots\cdots (12)$$

であり，Eは[R']のノルムを正規化するパラメータである。式(11)で得られる出力R'と望ましい出力Tとの差を次の式(13)で表わす。

$$R_{dif} = T - R' \qquad \cdots\cdots (13)$$

式(14)を使ってR_{dif}で更新学習することにより伝達関数の充実が図られる。

$$[\Delta X] = [S]^{H} R_{dif} \qquad \cdots\cdots (14)$$

ここで，ΔXはメモリXの更新量である。式(10)～式(13)を式(14)に代入すると

$$[\Delta X] = [S]^{H}([T] - \frac{1}{E}[S][X]) \qquad \cdots\cdots (15)$$

となる。式(15)の追加学習の計算過程で同式の右辺がゼロとなると，伝達関数[X]は厳密に式(7)で表される。式(15)に示すようにシナプスの更新量は学習誤差に比例するので，式(15)で得られる[X]の改善は最急降下法と同様速い。ここでは設計変数の陽な2次関数の最小値を探索するため，式(11)から式(15)までの反復は数回で終了する。ただし，実用上は当然のことながら収束のしきい値が与えられる(例えば0.2%)。式(11)を展開すると

$$r'_k e^{i\varphi'_k} = \frac{1}{E}\sum_{p=1}^{n} (x_p^k)\lambda'_p e^{i\theta'_p} \qquad \cdots\cdots (16)$$

となる。ここで，x_p^kは伝達関数[X]のp行k列成分である。式(16)に示すように任意の非線形関数は変換した関数族の線形の組み合せで表せる。例えば，フーリエ変換は，調和関数族の線形の組み合わせによって任意関数の近似化ができる。以上の追加学習によって任意の精度の学習が可能となることが保証されている。さて，HNNの学習能力は式(15)に示したメモリ[X]に関係し，その大きさは入力の数に依存する。情報量を増やすためBNNでは中間層のニューロンの数を増やすのに対して，HNNでは内部入力ﾏﾄﾘｸｽ[S]の成分を次式でﾏﾄﾘｸｽ[S]を大きくすることができる。つまり，外部入力数そのものは不変だが，既存の入力データを使って，これまでのものと線形関係でないことを保証するために入力データ同志の乗で新たな入力データを作成するものである。例えば，ニューロンへの基本的な入力を2とし，2次項を作成すると，次の拡張ができる。

$$S = [\lambda_1 e^{i\theta_1}, \lambda_2 e^{i\theta_2}]$$

$$\Downarrow \qquad \cdots\cdots (17)$$

$$\overline{S} = [\lambda_1 e^{i\theta_1}, \lambda_2 e^{i\theta_2}, \lambda_1 e^{-i\theta_1}, \lambda_2 e^{-i\theta_2}, \lambda_1^2 e^{i2\theta_1}, \lambda_2^2 e^{i2\theta_2}, \lambda_1^2 e^{-i2\theta_1}, \lambda_2^2 e^{-i2\theta_2}, \lambda_1\lambda_2 e^{i(\theta_1+\theta_2)}, \lambda_1\lambda_2 e^{i(-\theta_1+\theta_2)}, \lambda_1\lambda_2 e^{-i(\theta_1+\theta_2)}, \lambda_1\lambda_2 e^{i(\theta_1-\theta_2)}]$$

以上の操作で入力の数は2から12に増大する。

3. 関数近似を用いた最適値探索手順

設計領域を式(18)に示すX_0とする。

$$X_0 = [\{x_a\}, \{x_b\}] \qquad \cdots\cdots (18)$$

$$\{x_a\} = \{x_a^1, x_a^2, \ldots, x_a^n\} \qquad \cdots\cdots (19)$$

$$\{x_b\} = \{x_b^1, x_b^2, \ldots, x_b^n\} \qquad \cdots\cdots (20)$$

ここで，nは設計変数の数，x_a^i, x_b^i (i=1,...,n)はそれぞれ設計変数の下限，上限である。先ず，設計領域を次の式(21)で領域[0, 1]に変換する。

$$\{x'\} = \frac{\{x\} - \{x_a\}}{\{x_b\} - \{x_a\}} \in [\{0\}, \{1\}], \{x\} \in [\{x_a\}, \{x_b\}] \qquad \cdots\cdots (21)$$

ここで，$\{x'\}$は変換後の設計変数ベクトル，$\{x\}$は変換前の設計変数ベクトルである。これにより，設計領域はX_0からX_0'に変換される。X_0'を対象に領域内で最大値を求める最適値探索手順を次のようにする。

ステップ1. 先ず，全設計領域X_0'内を各設計変数に対して等分割し，分割の刻み幅をDx_0とする。

$$Dx_0 = \{dx_0^1, dx_0^2, \ldots, dx_0^n\} \qquad \cdots\cdots (22)$$

刻み幅の粗さは隣り合う分割区間，すなわち，連続する2つの区間内の極値（極大値及び極小値）点の数は高々1つとする。これらの分割点における関数値を算出し，NNで学習させる。学習したNNを基に極大値位置$\{x_{i0}\}$及び極大値$f(x_{i0})$を求める。ここで，$\{x_{i0}\} = \{x_{i0}^1, x_{i0}^2, \ldots, x_{i0}^n\}$ (i=1,2,...m)，mは極大値の数である。

ステップ2. 隣り合う分割区間内の極値の数が高々1つであってもステップ1での学習点数が少な過ぎると，予測した極大値位置並びに極大値の精度は十分でない可能性がある。そこで，次の操作で予測精度の向上を図る。

極大値を囲む分割領域内を再分割し，学習を行う。再分割領域を$[\{x_l\}, \{x_u\}]$とする。$\{x_l\}$及び$\{x_u\}$は次の式で得る。

$$\{x_l\} = \min(\{x_i\} - \{x_{i0}\}) + \{x_{i0}\} < \{x_{i0}\} \qquad \cdots (23)$$

$$\{x_u\} = \max(\{x_i\} - \{x_{i0}\}) + \{x_{i0}\} > \{x_{i0}\} \qquad \cdots (24)$$

ここで，$\{x_i\}$は$\{x_{i0}\}$と隣接する分割領域の節点座標である。極大値が節点上である場合を考慮し，式(23)と式(24)は完全不等式であることが必要である。そして，$\{x_l\}$と$\{x_{i0}\}$の中点及び$\{x_u\}$と$\{x_{i0}\}$の中点にそれぞれn点を追加して再学習を行う。その結果得られる極大値位置$\{x_{i1}\}$及び極大値$f(x_{i1})$を求める。

ステップ3. 終了条件を：

$$\max_{i=1,\ldots,m}\left(\frac{\|\{x_{i1}\} - \{x_{i0}\}\|}{\|\{x_{i0}\}\|}\right) < 5\% \qquad \cdots\cdots (25)$$

とする。式(25)が成立しない場合，同式が成立しないiに限り$\{x_{i0}\} = \{x_{i1}\}$，$f(x_{i0}) = f(x_{i1})$としてステップ2に戻し新たな学習点を追加して近似関数を求める。ただし，式(25)が成立す

るiに関しては，学習点の追加は行わない。全てのiに対して式(25)が成立するまでステップ2，3の繰り返しを行う。終了条件が満たされると，最大値は次式で得られる。

$$f(x_{k0}) = \max_{i=1,\cdots,m}\{f(x_{i0})\} \qquad \cdots\cdots (26)$$

4．数値解析例

4.1 多峰関数　まず，2次元のGaussian分布関数で作成された多峰関数を図1に示す。同図に示すように設計領域 x_i=[-2.0, 2.0] に関数値の峰が3つある。最大峰の位置は[0.000, 1.586]で，関数値は8.077であり，該当する所に丸が付されている。第3章に示す最適設計手順のステップ1を基に得られた分割を同図に示す。すなわち，設計領域において各々5等分割すれば連続する2つの区間内に存在する極値点の数は1か0となる。これら25の節点座標を入力とし，節点における関数値を出力としてHNNで学習させる。学習の収束のしきい値を0.2%とする。ニューロンの内部入力数を10から始め数回の繰り返しで誤差の最小値が得られても0.2%以内にならない場合がある。これは内部入力の数が少ないためであり式(?)に基づいて2ずつ入力数を増やした所，入力数30で5回の繰り返しで0.2%内の収束値が得られた。この学習結果をもとにHNNで算出した3つの極大値を図1及び表1に示す。図1，表1では第1から第3まで正しい極大値の大きさの順に番号をつけている。同図で"+"は正しい極大値の位置，"-"は正しい極小値の位置を示し，"0"はステップ1で求められた極大値位置を示す。表1で小括弧内の数値は極値のそれぞれの座標値を示している。次にステップ2に移り，いずれの極大点においてもステップ2の式(23)，(24)に基づき再分割し学習点数を増加する。この2回目の学習の追加で得られる極大値の位置のうち第1と第3番目は式(27)を満たした。従って第2番目の極大値点を対象に再度ステップ2の操作を行って極大点を求め直した。その結果，第1～第3の極大値はともに収束した。すなわちステップ2及び3の繰り返しは3回で終了した。その結果を表1の最終結果として示している。以上のループで合せて52点の解析値を使って全領域における最大値が求められた。最大値の予測誤差は0.2%である。

次に，HNNと比較するためにBNNで同一の学習条件で学習させ同じく3つの極大点の予測を行う。なお，使用するBNNの学習法は，通常の最急降下法ではなく，ローパスフィルタ性能を有する学習法[9]である。これは最急降下法などを使用する場合に比しより高速な学習が得られるとされている[10]。学習は中間層数は1つで，中間層ニューロン数を10から始め8000回の繰り返しで誤差の最小値が求めても0.2%以内にならない場合中間増のニューロンの数が少ないことを意味するので2ずつ中間増ニューロン

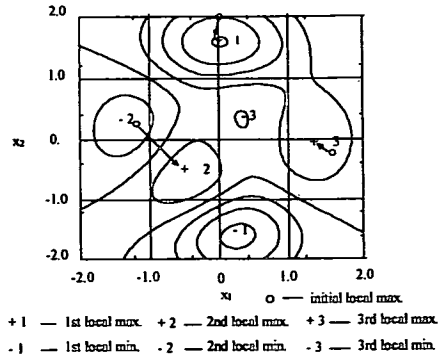

Fig.1. Dividing method of designed space for multiple peaks function

Table 1. Local maximum and optimized results

	1st local max.	2nd local max.	3rd local max.
initial by HNN	5.61 (0.00,2.00)	3.82 (-1.17,0.35)	3.85 (1.45,-0.21)
final by HNN	8.06 (-0.07,1.59)	3.78 (-0.48,-0.62)	3.56 (1.31,-0.08)
initial by BNN	5.90 (0.051,1.90)	0.33 (-2.00,-1.49)	2.16 (1.08,0.30)
final by BNN	7.21 (0.08,1.56)	3.99 (-0.48,-0.64)	3.57 (1.24,0.04)
true result	8.08 (-0.00,1.59)	3.77 (-0.50,-0.62)	3.57 (1.31,-0.07)

数を増やした所，入力数20で498回の繰り返しで0.2%内の収束値が得られた。この学習結果をもとにステップ1によりBNNで算出した3つの極大値を表1に示す。次にHNNと同様に3点ともステップ2の式(23)，(24)に基づき再分割した。この2回目の学習の追加で得られる最大値のうち第1極大点では式(26)を満たすので第2，第3の極大点でのみ再分割し，極大点を求め直した。その結果，第1，第2点は収束した。次に第3番目の極大点に対してのみ再分割し，再度の学習ですべての極大点は収束した。つまり，4回目で収束し，解析値点数は計61点である。その結果は表1のBNNの最終結果として示している。結局HNNより，9点解析値を多くする必要があった。そして求めるべき最大値の誤差は10.8%であり，HNNの誤差0.2%と比べるとその精度は格段に劣る。BNNの場合，中間層のニューロンの数を適切に選ぶに際してHNNの場合は[S]の成分数を適切なものにするために共に繰り返し計算を行うがこの過程でBNNの場合に必要な計算時間はHNNに比べて圧倒的に長い。

4.2 簡易衝突モデル　本研究では車両の衝突問題の最適化解析を目指しているが，ここでは車両を模擬した図2に示す2自由度のばね-マスモデルを扱う。同図で，$k_1(x)$はキャブとフロント間のばね，$k_2(x)$はフロントのばねで共に非線形である。それらは，例えば式(27)のように表される。

$$k_i(x) = s_i * x^a \qquad (i=1, 2) \qquad \cdots\cdots\cdots\cdots (27)$$

ここで，s_i(i=1,2)は材料に関係なく構造の寸法に関係する定数で，a は材料に関係する定数である。a の値により系の非線形の強さを変えることができる。例えば，a=1 のとき，ばねは完全な線形ばねであり，a<1 の場合は，軟ばねである。つまり，変曲点，すなわち，降伏点後ばねの荷重の増加量はばねの変形量の増加に伴って少なくなって行く。 a>1 の場合は，硬ばねで降伏点後ばねの変形量の増大に伴って荷重の増加量は大きくなって行く。用いたモデルの質量は m_1=300kg, m_2=150kg，初速度は V_0=20m/s である。ここでフロントエンド部すなわち $k_2(x)$のｴﾈﾙｷﾞｰ吸収量を最大にする，$k_1(x)$，$k_2(x)$の組み合せを求める。学習のための $k_2(x)$のｴﾈﾙｷﾞｰ吸収量の算出はﾆｭｰﾏｰｸ β 法[11]を使用し，β=1/4 としている。非線形性を強くして a=0.50 とすると，設計変数を 10^4～10^5 とする範囲では，ばね $k_2(x)$のｴﾈﾙｷﾞｰ吸収関数は図 3 のようになる。NN の学習に用いたパラメータの組み合せは図 3 に示すように s_1, s_2 をいずれも 5 等分割した 25 通りのものである。同図に示すようにこの領域内には 3 つの極大値と 1 つの極小値があるが，図 3 の分割は第 3 章のステップ 1 の分割条件を満たしている。同図での数字の付け方，+，－，○の意味は図 1 と同様である。以上の準備のもとに第 3 章のステップ 1 の操作に入る。まず，これらの 25 通りの設計変数の組み合わせの座標値を，NN の入力として，対応する $k_2(x)$の吸収ｴﾈﾙｷﾞｰを NN の出力として学習を行う。ここで，学習の収束のしきい値を 0.2%とする。ニューロンの内部入力数を 5 から始め式(17)に基づいて 5 ずつ入力数を増やした所，入力数 35 で 0.2%内の収束値が得られた。この間に要した学習回数は 56 回であり，学習時間は約 49 秒（pentium-pro180）である。ステップ 1 で予測した各極大値及びその極大点の位置を表 2 に示す。次にステップ 2 に移り，3 つの極大値位置のそれぞれの近傍に学習点を追加し，再度学習を行う。この 1 回目の追加学習ではニューロンの内部入力数を 35 からスタートして 38 で学習条件を満たした。これによって求められた 3 つの極大点のうち第 3 極大値のみ式(26)を満たした。従って第 1，第 2 の極大点に対して再分割し，極大点を求め直した。この 2 回目の学習でニューロンの内部入力数を 38 からスタートして 40 で学習条件を満たした。この 2 回目の学習ですべての極大点は学習させ同じく 3 つの極大点の予測を行う。中間層数は 1 つで，中間層ニューロン数を 2 から始め，8000 回繰り返してもしきい値 0.2%以内とならなければ，式(26)を満たして収束した。その結果を表 2 に示す。最大点位置の誤差は 5.3%である。解析値点数は計 44 である。この間の学習時間は約 70 秒である。さて，この最適結果はフロントばねの強さは制限内の最大値で，キャブのばねは最小値と明確な結果であり，キャブの相対変位は 0 と理想的な潰れモードとなった。このように 2 つのばねが明確な値となったのはばねに対する重量などの条件を加えなかったからである。実際の設計では 2 つのばねの重量の和が一定以下というような条件が加えられる。この場合も今回示した手順はそのまま利用できる。

次に，HNN と比較するために BNN で同一の学習条件でニューロン数を 2 つずつ増加させ 10 まで増やして 4500 回で収束した。この間に要した学習回数は(4x8000+4500) 回で計算時間は約 4500 秒である。以上 BNN に要した学習時間は HNN の 60 倍である。ここで，使用する BNN は第 4.1 節で使用したのと同じ文献(11)に従うものである。次に第 3 章のステップ 2 に移り，3 つの極大値近傍に学習点を追加し，再学習を行う。ニューロンシナプスの結合強さ及びバイアス値はステップ 1 の学習の収束時のものとしたが中間層ニューロンの数 10 では収束しなかった。そこで，中間層ニューロンの数を 12 に増やす。増やしたニューロンに関わるシナプスの結合強さ及びバイアス値の初期値のみ乱数で与え，学習を試みたが，8000 回の繰り返しで誤差は 0.2%以内に収まらなかった。従って更にニューロンの数を増加させる必要がある。しかし，ニューロンの数 10 及び 12 の場合に要した CPU 時間は 5000 秒である。この時点で既に HNN の場合の 70 倍の CPU タイムを要している。従って本解析における HNN の BNN に対する優位性は明白である。

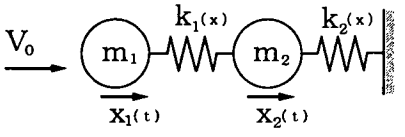

Fig.2 Mathematical model for shock analysis

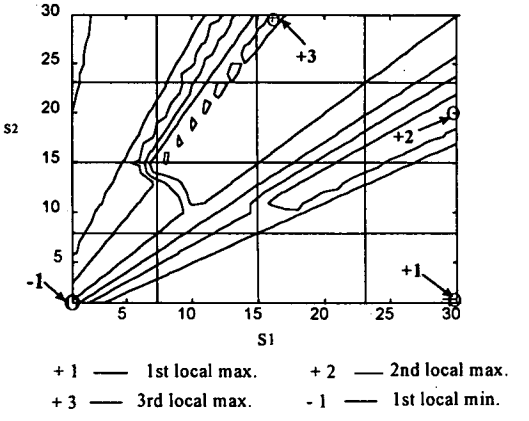

Fig.3 Dividing method of energy dissipation of k_2 (a=0.50) for training

Table 2. Local maximum and optimized result

	1st local max.	2nd local max.	3rd local max.
initial result	9.32×10^4 (0.84, 0.05)	8.37×10^4 (0.93, 0.53)	6.87×10^4 (0.50, 1.00)
final result	9.21×10^4 (0.95, 0.02)	8.37×10^4 (1.00, 0.63)	7.19×10^4 (0.55, 1.00)
true result	8.98×10^4 (1.00, 0.00)	8.29×10^4 (1.00, 0.63)	7.12×10^4 (0.58, 1.00)

以上のように，学習誤差を目的の値以下にするために，HNNでは適切な内部入力の数が，BNNでは中間層のニューロンの数が求められる。その過程はいずれも最適化技術が援用されるが，HNNの誤差関数は設計変数の陽な2次関数であり，その最小値を求める過程，すなわち内部入力数が適切かの確認は数回の反復で終了する。一方，BNNの誤差関数は設計変数の陰関数となるため，中間層のニューロンの数が適切かの確認は一般に数千回の反復を要する。ここでの例題でのHNNの優位さはこの点も1つの大きな理由となっている。

5. まとめ及び今後の課題

(1) 非線形構造への最適化技術の援用が強く望まれており，NNの近似能力を利用して目的関数の関数形状そのものを近似し最適値を得る方法が考えられ，それを実現するための手順の開発を行った。

(2) 構造最適化の分野ではHNNの利用の試みはなされていないため，特に，写像関数の選択や入力の拡張法が明確となるようBNNと比較する形でその理論，特性を整理した。

(3) すなわち，a) 学習は簡単な基底関数の陽な2次関数の最小値を探索することから線形の組み合せの係数[X]を決めることに帰着される。b) 初期値を決める難しさがなくなるので，学習が速く局所最小値に陥る可能性はない．c) 入力同士の乗算を新たな入力とすることにより学習能力を高めることができる。

(4) 関数が非線形となると無条件でその形状を近似するのは容易でない。そこで，解析者が極値の数とその位置をほぼ把握しているという条件下で，解析回数を極力少なくするため，必要な初期分割程度とその後に続く関数の形状を近似する手順を示した。

(5) 今回提示した最適手順を多峰関数及び前面衝突を模擬した2自由度非線形ばねを有する衝突モデルに適用した。

(6) いずれの例題においても，BNNよりHNNの方がはるかに短い時間で近似できることを示した。

今後の課題として次のことがあげられる。

(1) 今回，極値の場所がおおよそ分かっているとし，全体を近似するNNの構築を行ったがこの他に個々の極値の近傍だけをカバーする複数のNNを構築する方法が考えられる。両手法の得失に関する議論。

(2) 解析者が極値の数と位置が分からない，より一般的な問題について検討できる手順の開発。

(3) 今回，開発した手法は大規模な問題にもそのまま適用できるものであるがそれの実際のモデルでの確認。

謝辞

本研究はJR東日本寄附講座「協調工学講座」の研究の一環として行なった研究であり，また，文部省科学研究費の援助も受けて実施されたことを記して関係者各位に謝意を表する。

参考文献

(1) 萩原，耐衝突強度設計への応用とその自動車構造設計への応用，日本機械学会材力部門講習会‘強度設計における有限要素法の基礎から応用まで’，1994年12月。

(2) J. Fukushima et al., Shape Optimization for Impact and Crash Problem of Car Bodies Using DYNA3D, 2nd U.S. National Congress on Computational Mechanics, USACM, 1993

(3) S.Kodiyalam and R. Gurumoorthy, Neural Networks with Modified Backpropagation Learning Applied to Structural Optimization, AIAA Vol.34, No.2 Feb. 1996

(4) Vito Leonardo Plantamura, et al., The Brain Windows in Evoked Potentials and the Holographic Network for Neurological Diagnoses, Proc. International Joint Conference on Neural Networks, Nagoya, Japan, 1994

(5) Vito Leonardo Plantamura, et al., The Holographic Fuzzy Learning for Credit Scoring, Proc. International Joint Conference on Neural Networks, Nagoya, Japan, 1994

(6) J.G.Sutherland, The Holographic Model of Memory, Learning and Expression, International Journal of Neural System, 1990, Vol. 1 No.3, pp.259-267

(7) 舟橋，ニューラルネットワークのcapabilityについて，信学技報，MBE-88-52，1988.

(8) W.C.Carpenter and J.F.M. barthelemy, A Comparison of Polynomial Approximation and Artificial Neural Net as Response Surface. AIAA/ASME/ASCE/AHS 33rd Conf,1992

(9) R.Reed, Pruning Algorithms - A Survey, IEEE Trans., Vol.NN-4, no.5,pp.704-747,1993

(10) T. P. Vogl, etc., Accelerating the convergence of the backpropagation method, Biological Cybernetics, vol.59,1988,pp.257-263

(11) N. M. Newmark, A Method of Computation for Structural Dynamics, Journal of Eng. Mech. Div., ASCE,Vol.85, No. EM3, Proc. Paper 2094, July 1959, pp.67-94