

I-256

最適設計におけるAnalyzerとしてのニューラルネットワーク

八戸工業高等専門学校 正員 ○齊藤 進
早稲田大学理工学部 正員 堀井健一郎
早稲田大学理工学部 正員 依田 照彦

1. まえがき

近年,人間の神経細胞(ニューロン)のモデルをネットワーク化し,脳の中の情報処理を模擬したニューラルネットワークの研究・応用が,交通計画,損傷評価,景観評価,データ解析など多方面において試みられつつある。最適設計の分野においてもニューラルネットワークは,①Analyzerとしての応用,②Optimizerとしての応用,③ニューラルネットワークのみによる最適設計 などの利用が可能である。このうち本研究では,“Analyzerとしてのニューラルネットワーク”を組み込んだ最適設計を試み,Analyzerとしての応用についてその有効性,問題点などを明らかにしたものである。

2. 階層型ニューラルネットワークと学習

階層型ニューラルネットワークは,人工ニューロン(ユニット)を図-1に示すような階層型のネットワークとして結合したものである。ひとつのユニットは,多入力出力の素子であり他のユニットとは神経繊維を模擬した結合の強さを表す結合荷重 $w_{ij}^{(m)}$ で結ばれている。図-1における $y_i^{(m)}$ は第 m 層($m=0, 1, \dots, M$)の i 番目のユニットの出力を示し,次の(1)式~(3)式によって表される。

$$x_i^{(m)} = \sum_j w_{ij}^{(m-1)} y_j^{(m-1)} - \theta_i \quad \dots (1)$$

ただし,しきい値 θ_i は1を出力する $y_0^{(m-1)}$ の結合荷重 $w_{i0}^{(m)}$ として処理する。 $x_i^{(m)}$ は出力関数(シグモイド関数),

$$F[x] = 1 / (1 + e^{-x}) \quad \dots (2)$$

による変換を受け, $y_i^{(m)} = F[x_i^{(m)}]$ $\dots (3)$

がユニットから出力される。図-1の第0層(入力層)のユニットに外部からデータが入力されれば,(1)式~(3)式に従って出力層に向かって順次各層のユニット出力が計算され,最後に第 M 層(出力層)のユニットよりネットワークとしての出力がなされる(順伝播)。学習データとして一組の入力データおよび期待出力(教師データ)が準備されている時,ネットワークの出力が期待出力に近づくように,結合荷重 w_{ij} を更新していくことを“学習”という。学習はBP法によって行う。

階層型ニューラルネットワークは,以下のような能力的特質をもつためその応用範囲が広い。

- (a) “学習”によって多入力(設計変数)と多出力(制約関数や目的関数)の間で非線形写像が可能になる。
- (b) 学習済みネットワークは未学習の入力に対しても適正な値を迅速に出力できる。(汎化能力)
- (c) 学習済みネットワークによって,出力の入力に対する感度係数の計算が迅速にできる。

本研究では上の(a),(b),(c)の特質を利用して,目標計画法を用いた従来の多目標構造最適設計に“Analyzerとしてのニューラルネットワーク”を組み込んだ方法を試みる。

3. 感度係数の計算方法

多目標構造最適設計に目標計画法を適用しこれを逐次線形化法で解く場合には,設計変数の制約条件および目的関数に対する偏微分係数すなわち感度係数の計算が必要となるが,学習済みの階層型ニューラルネットワークを用いれば,簡単な積和演算のみで感度係数の計算が可能になる。ここでは中間層が2つの4層のネットワークを対象に感度係数の計算方法を考える。いま,入力層,第1中間層,第2中間層,出力層のユニットを代表する添字を, l, k, j, i , ユニット出力を y_l, y_k, y_j, y_i , ユニット出力関数を F_k, F_j, F_i で表すと,(1)式,(3)式より出力とその偏微分は,第1中間層,第2中間層,出力層について,

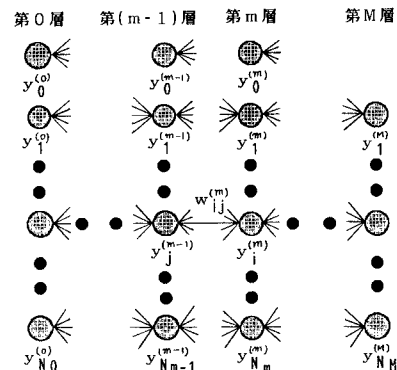
$$x_k = \sum_l w_{kl} y_l, y_k = F_k[x_k], \partial y_k / \partial y_l = (\partial F_k / \partial x_k) (\partial x_k / \partial y_l) = F'_k[x_k] w_{kl} \quad \dots (4)$$


図-1 階層型ニューラルネットワーク

$$x_j = \sum_k w_{jk} y_k, y_j = F_j [x_j], \partial y_j / \partial y_k = (\partial F_j / \partial x_j) (\partial x_j / \partial y_k) = F'_j [x_j] \sum_k w_{jk} \dots (5)$$

$$x_i = \sum_j w_{ij} y_j, y_i = F_i [x_i], \partial y_i / \partial y_j = (\partial F_i / \partial x_i) (\partial x_i / \partial y_j) = F'_i [x_i] \sum_j w_{ij} \dots (6)$$

従って、出力 y_i の入力 y_i に関する感度係数は、

$$\partial y_i / \partial y_i = (\partial y_i / \partial y_j) (\partial y_j / \partial y_k) (\partial y_k / \partial y_i) \dots (7)$$

上式に(4)式、(5)式、(6)式の偏微分を代入し、

$$\partial y_i / \partial y_i = (F'_i [x_i] \sum_j w_{ij}) (F'_j [x_j] \sum_k w_{jk}) (F'_k [x_k] w_{ki}) \dots (8)$$

いま、 F_k 、 F_j 、 F_i をすべて(2)式のシグモイド関数とすると、

$$\partial y_i / \partial y_i = [\varepsilon y_i (1 - y_i) \sum_j w_{ij}] [\varepsilon y_j (1 - y_j) \sum_k w_{jk}] [\varepsilon y_k (1 - y_k) w_{ki}] \dots (9)$$

4. 目標計画法による多目標構造最適設計

非線形目標計画法による多目標構造最適設計の制約条件、目的関数、それらの感度係数の計算部分を、学習済みニューラルネットワークの利用で置き換えると、“ニューラルネットワークをAnalyzerとして用いた多目標構造最適設計”となる。ニューラルネットワークの学習には一定の時間がかかるものの、一度学習済みとなったネットワークによる関数計算（汎化）や感度係数の計算は、(1)式～(3)式の順伝播と(9)式の積和演算のみであるから最適設計自体は極めて高速な計算となる。

5. トラスの多目標最適設計への応用

図-2に示す20部材トラスの多目標最適設計を考える。設計変数は部材断面積 $A_1 \sim A_5$ （断面は正方形Box断面）、制約条件は先端の水平変位 D について、 $D \leq D^*$ （ D^* は許容値）、目的関数は面内全体座屈荷重係数 B と重量 W とする。ニューラルネットワークとしては1出力モジュール型 $N(5-10-1)$ を3つ用い、各々を D 、 B 、 W に対応させる。 $A_1 \sim A_5$ に対して各々等間隔に5つの断面積（25, 30, 35, 40, 45 cm^2 ）を考えると3125通りの組み合わせとなるが、これからランダムに100個の組み合わせを選び、対応する D 、 B 、 W を計算して学習データとする。学習データは入力・出力ともにすべて0.1～0.9の範囲にスケリングをしてから学習を行い、出力と期待出力（教師データ）の2乗誤差の平均が0.00005以下になった時を収束とみなす。学習済みニューラルネットワークによる関数値と感度係数の計算結果の一例を表-1に示す。

多目標最適設計の例としては以下の3ケースを考え、Analyzerとしてのニューラルネットワークを用いた場合と従来法による場合について計算を行った。

(a) $D \leq 0.8\text{cm}$ のもので W を最小にする。

(b) $D \leq 0.8\text{cm}$ のもので $W \leq 4290\text{kgf}$ 、次に B を最大化。

(c) $D \leq 0.8\text{cm}$ のもので $B \geq 6.00$ 、次に W を最小化。

結果は表-2に示すが、これから分かるように両者にはほとんど差がなく、Analyzerとしてのニューラルネットワークの有効性が明かである。

6. あとがき

以上、学習済みニューラルネットワークによって、関数値と感度係数の計算が極めて簡単かつ正確にでき、ニューラルネットワークをAnalyzerとして利用できることを明らかにした。しかし、学習データの準備とニューラルネットワークの学習にはかなりの時間がかかりこれをいかに速めるかが残された問題である。

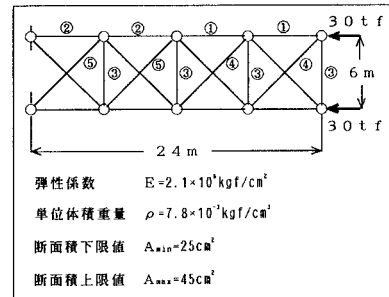


図-2 20部材トラスと計算のための諸元

表-1 関数値と感度係数についての比較の一例

設計変数 (cm^2) $A_1=30, A_2=35, A_3=28, A_4=40, A_5=33$						
$D(\text{cm})$	$\partial D / \partial A_1$	$\partial D / \partial A_2$	$\partial D / \partial A_3$	$\partial D / \partial A_4$	$\partial D / \partial A_5$	
0.869	0.0124	0.0101	0.0025	0.0015	0.0014	
0.861	0.0121	0.0096	0.0024	0.0013	0.0013	
B	$\partial B / \partial A_1$	$\partial B / \partial A_2$	$\partial B / \partial A_3$	$\partial B / \partial A_4$	$\partial B / \partial A_5$	
3.115	0.0326	0.1027	0.0777	0.0191	0.0540	
3.098	0.0155	0.1080	0.0712	0.0194	0.0679	
$W(\text{kgf})$	$\partial W / \partial A_1$	$\partial W / \partial A_2$	$\partial W / \partial A_3$	$\partial W / \partial A_4$	$\partial W / \partial A_5$	
3647	2497	2566	2623	3699	3722	
3673	2400	2400	2400	3394	3394	

（上段がニューラルネットワーク値、下段が理論値）

表-2 最適設計の結果

ケース	制約・目的関数値			設計変数 (cm^2)				
	$D(\text{cm})$	B	$W(\text{kgf})$	A_1	A_2	A_3	A_4	A_5
(a)	0.800	2.650	3180	37.4	37.0	25.0	25.0	25.0
	0.801	2.709	3183	37.8	36.5	25.0	25.0	25.0
(b)	0.655	7.172	4280	44.2	45.0	45.0	27.0	39.1
	0.714	6.830	4281	35.7	43.4	45.0	33.0	41.0
(c)	0.692	5.998	3840	40.6	45.0	45.0	25.0	27.0
	0.688	6.002	3851	40.8	45.0	45.0	25.9	27.2

（上段がニューラルネットワーク値、下段が従来法による値）