3Ｋ－3

バイナリニューラルネットワークによる多値情報処理

上久保　浩　　　黒川　恭一
防衛大学校情報工学教室

1. はじめに

出力値として2値状態をとるバイナリニューロン[1]を用いた相互結合型ニューラルネットワークによる種々の組み合わせ問題の解法が多数提案されている[2]。本稿において扱う輸送問題[3]についても、土村等が1つのバイナリニューロンを単位輸送量に対応させるネットワークを提案している[4]。しかし、この解法においては、輸送量の増加に伴いニューロン数が線形に増加するという問題点がある。本稿においては、この輸送問題に対し、上記の問題点解決の1手法として、多値化されたバイナリニューラルネットワークを提案し、その有効性を検証する。

2. 輸送問題の概要

輸送問題には、制約条件の種類により様々な型が有るが、本稿で扱う問題は、Hitchcock型の輸送問題と呼ばれるものである。これは、ORの分野で良く知られており、以下のように定義される。

1. ある商品に関して、m個の供給地S_i $(i=1,2,\cdots,m)$と、n個の需要地D_j $(j=1,2,\cdots,n)$があり、S_iでの供給量はs_i、D_jでの需要量はd_jとする(図1)。但し、総供給量と総需要量は等しい。
$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j \tag{1}$$
2. また、供給地S_iから需要地D_jへの輸送量x_{ij}を$(i=1,2,\cdots,m, j=1,2,\cdots,n)$、商品一個当たりの輸送コスト$c_{ij}$を$(i=1,2,\cdots,m, j=1,2,\cdots,n)$とする。
3. 以上の条件下で、全体の輸送コストを最小にする。

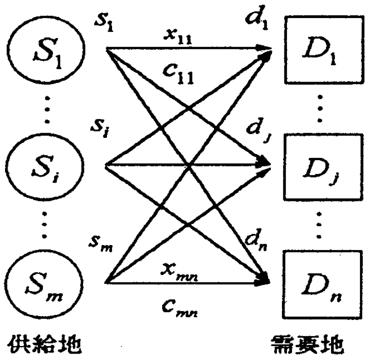

図1:輸送問題

この問題は一般に、
$$\text{制約条件}:\sum_{j=1}^{n} x_{ij} = s_i,\ \sum_{i=1}^{m} x_{ij} = d_j \tag{2}$$
のもとで、
$$\text{目的関数}:C = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{3}$$
を最小にする線形計画問題として定式化される。

3. ニューラルネットワークによる輸送問題の解法

3.1. 輸送問題のニューラルネットワーク表現

文献[4]の解法では、各バイナリニューロンが1単位の輸送量に対応している。そのため、1つの供給地から、1つの需要地への輸送量の考えられる最大値x_{max}は、
$$x_{max} = \min\{\max\{s_i\},\max\{d_j\}\} \tag{4}$$
となり、この問題を表現するためには、$m\times n\times\ x_{max}$個のバイナリニューロンより成る3次元のバイナリニューロンアレイが必要になる。

本稿で提案する解法においては、バイナリニューロンペア(以下、BNPと略記)を2次元に配置した$m\times n$個のバイナリニューロンより成るバイナリニューロンペアアレイ(以下、BNPAと略記)を用いる。ここでBNPは、2つのバイナリニューロンBN^+とBN^-を対にしたものであり、これらの出力を輸送量の変化に対応させる。具体的には、図2に示すような輸送量を増加させる方向に働くバイナリニューロンBN^+と減少させる方向に働くバイナリニューロンBN^-である。カウンタに輸送量x_{ij}を保持させ、これをBN^+_{ij}とBN^-_{ij}の発火によりアップあるいはダウンカウントさせるものである。そのため、BN^+_{ij}、BN^-_{ij}の出力V^+_{ij}、V^-_{ij}と輸送量x_{ij}の変化量dx_{ij}との関係は、以下の様になる。
$$x_{ij}(t+1) = x_{ij}(t) + dx_{ij} \tag{5}$$
$$dx_{ij} = \begin{cases} 1 & (V^+_{ij}=1, V^-_{ij}=0) \\ 0 & (V^+_{ij}=V^-_{ij}) \\ -1 & (V^+_{ij}=0, V^-_{ij}=1) \end{cases} \tag{6}$$

これにより、バイナリニューロンの数は、$m\times n\times 2$個に抑えられる。

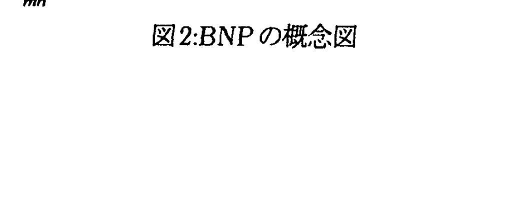

図2:BNPの概念図

Multivalue Processing Using A Binary Neural Network
Hiroshi Kamikubo, Takakazu Kurokawa
Department of Computer Science, National Defense Academy
1-10-20 Hashirimizu,Yokosuka,Kanagawa 239,Japan

3.2. 入出力関数及び動作式

本提案においては、バイナリニューロンを用いたが、その入出力関数は以下の様になっている。

$$V_{ij}^{+}=\begin{cases}1 & U_{ij}^{+}>0\\ 0 & U_{ij}^{+}\le 0\end{cases} \tag{7}$$

$$V_{ij}^{-}=\begin{cases}1 & U_{ij}^{-}>0\\ 0 & U_{ij}^{-}\le 0\end{cases} \tag{8}$$

また、動作式は以下のようになる。

$$\begin{aligned}\frac{dU_{ij}^{+}}{dt}=&A^{+}\left\{\left(s_i-\sum_{l=1}^{n}x_{il}\right)+\left(d_j-\sum_{k=1}^{m}x_{kj}\right)\right\}\\&+B^{+}(c_{max}-c_{ij})\\&+C^{+}\left\{\left(\sum_{l=1}^{n}(c_{il}-c_{ij})\right)+\left(\sum_{k=1}^{m}(c_{kj}-c_{ij})\right)\right\}\\&-D^{+}V_{ij}^{-}\end{aligned} \tag{9}$$

$$\begin{aligned}\frac{dU_{ij}^{-}}{dt}=&A^{-}\left\{\left(\sum_{l=1}^{n}x_{il}-s_i\right)+\left(\sum_{k=1}^{m}x_{kj}-d_j\right)\right\}\\&+B^{-}(c_{ij}-c_{min})\\&+C^{-}\left\{\left(\sum_{l=1}^{n}(c_{ij}-c_{il})\right)+\left(\sum_{k=1}^{m}(c_{ij}-c_{kj})\right)\right\}\\&-D^{-}V_{ij}^{+}\end{aligned} \tag{10}$$

但し $c_{max}=\max\{c_{ij}\}$, $c_{min}=\min\{c_{ij}\}$ である。

式(9)がBN^{+}の動作式である。第1項は供給量、需要量に満たなければ発火、超過していれば未発火にさせる項である。第2項はコストの低いものを発火させる興奮性の項である。第3項は同一供給地、及び同一需要地で、他よりコストが低ければ発火させる興奮性の項である。第4項は、BN^{-}が発火していれば、自身を未発火にする抑制性の項である。

一方、式(10)がBN^{-}の動作式であり、この式は式(9)と対称的に働く。

4. アルゴリズム

上記の動作式に従ったシミュレーションでは、以下のアルゴリズムに従った。尚、動作式の近似には、1次のオイラー法を用いた。

Step1: $t:=0$

Step2: $U_{ij}^{+}(0),U_{ij}^{-}(0)$を$-U_0\le U_{ij}(0)\le 0$の間の任意の値に初期化。

Step3: $V_{ij}^{+}(0),V_{ij}^{-}(0)=0$に初期化。

Step4: $x_{ij}(0)=0$とする。

Step5: $t:=t+1$

Step6: $U_{ij}^{+}(t),U_{ij}^{-}(t)$を式(9),(10)により更新。

Step7: $V_{ij}^{+}(t),V_{ij}^{-}(t)$を式(7),(8)により更新。

Step8: 全ての$V_{ij}^{+}(t),V_{ij}^{-}(t)$に変化が無く、かつ、$X_{ij}(t)$が解としての条件を満たしていれば解が求められたとして終了。

Step9: $t=T$(Tは事前に設定)ならば打ち切り。

Step10: 8,9以外であれば、Step5へ戻る。

5. シミュレーション結果

上記アルゴリズムにより、ソフトウェアシミュレータをワークステーション上に作成した。例として、$m=5,n=5$のサイズの表1のような輸送問題に関して1000回の試行を行い、正常に動作することが確認された。また、その試行の結果得られた解の輸送コストとその出現度数を、本稿で提案するアルゴリズムによるものと、文献[4]のアルゴリズムによるものとの比較を図3に示す。ここで、本稿で提案するアルゴリズムによるシミュレーションでは、式(9),(10)の各係数はそれぞれ、$A^{+}=A^{-}=200,B^{+}=B^{-}=1,C^{+}=C^{-}=1,D^{+}=D^{-}=1$とした。

図3より、本稿で提案するアルゴリズムは、文献[4]のアルゴリズムに比べ、最適解の出現率が高いことがわかる。

表1:例題

c_{ij}		需要地 D_j 1	2	3	4	5	s_i
供給地 S_i	1	11	15	3	15	1	3
	2	3	1	15	7	8	7
	3	12	4	8	11	13	3
	4	9	4	14	1	12	12
	5	10	14	3	9	1	13
d_j		11	14	3	9	1	

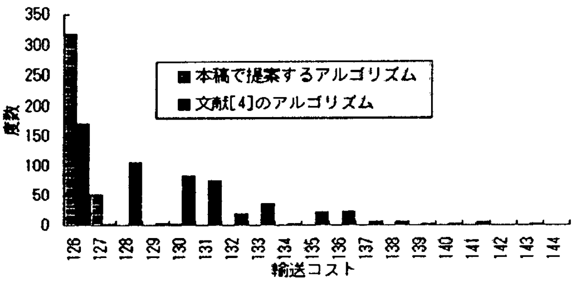

図3:輸送コストと出現度数

6. むすび

本稿では、多値情報を扱う必要のある問題例として輸送問題を挙げ、その一解法としてバイナリニューロンを輸送量の変化に対応させる方式を提案した。この解法は、ニューラルネットワークによる従来の解法に比べ少ないニューロン数で問題を表現でき、効果的に最適解を求めることができることがわかった。今後、より大きなサイズの問題への適用を図る。

参考文献

[1] McCulloch and Pitts:“A logical calculus of the ideas immannent in nervous activity”,Bulletin of Mathematics and Biophysics,5(1943).

[2] Y.Takefuji:“Neural Network Parallel Computing”, Kluwer Academic Publishers(1992).

[3] 伊理正夫:“線形計画法”,共立出版(1986).

[4] 土村,狐塚,黒川,“バイナリニューロンによるニューラルネットワークを用いた輸送問題の並列解法”,1992年電子情報通信学会春季大会,pp.6-42(1992).