

ホップフィールドニューラルネットワークによる 空間的に一様でないセルラニューラルネットワークの設計法

上田 康弘[†] 川原 雅一[†] 井上 貴志[†] 西尾 芳文[†]

[†] 徳島大学 〒770-8506 徳島市 南常三島町 2-1

E-mail: †{yasuhiro,kawahara,nishio}@ee.tokushima-u.ac.jp

あらまし 本研究では、ホップフィールドニューラルネットワーク (Hopfield NN) の連想記憶の機能を用いた、空間的に一様でないセルラニューラルネットワーク (CNN) の設計法を提案する。CNN はテンプレートの分布から 2 種類に分類できる。“空間的に一様な CNN”は実装コストが安価であることから、盛んにそのアプリケーションの研究がなされているが、一度に 1 つの処理しか実行できない。一方、“空間的に一様でない CNN”は高度な処理を実行できるが、設計法は複雑である。しかし本手法は連想記憶を用いて適材適所にテンプレートを配置するため“空間的に一様でない CNN”を簡単に実現できる。このシステムにおいて、二値画像を用いたシミュレーションを行い、その効果を確認する。

キーワード セルラニューラルネットワーク, ホップフィールドニューラルネットワーク, 画像処理

Design Method of Space-Varying Cellular Neural Networks by Using Hopfield Neural Networks

Yasuhiro UEDA[†], Masakazu KAWAHARA[†], Takashi INOUE[†], and Yoshifumi NISHIO[†]

[†] Dept. of Electrical and Electric Engineering Tokushima University

2-1, Minami-Josanjima, Tokushima 770-8506, Japan

E-mail: †{yasuhiro,kawahara,nishio}@ee.tokushima-u.ac.jp

Abstract In this study, we propose a design method of space-varying cellular neural networks (CNN) by using Hopfield neural network (Hopfield NN). CNN is classified into two types of systems like space-invariant system and space-varying system. Space-invariant and space-varying mean that all cells have identical template and not identical templates, respectively. The processing of the former is simple. On the other hand, the latter carries out complicated processing. Therefore, the design method is not easy. However, our proposed method is easy because we need only some existing templates and the associated patterns. Some simulation results show the basic properties of the proposed CNN and its effectiveness.

Key words Cellular Neural Networks, Hopfield Neural Networks, Image Processing

1. ま え が き

現在、様々な分野において画像処理の技術が用いられているが、年々処理すべき情報量が増え、さらに高度かつ高速な処理を行う技術の開発は急務である。しかし、現在一般的に用いられているデジタル処理は逐次演算を行うため、処理に時間が掛かってしまう。そこで人間の神経系を基に考案された様々なニューラルネットワークを用いた情報処理が盛んに行われ、既に各方面での実用化が進んでいる。その応用は連想メモリ、パターン認識、画像処理など幅広いものがある。さらにニューラルネットワークとセルラオートマトンの概念を組み合

わせたニューラルネットワーク、セルラニューラルネットワーク (Cellular Neural Networks: CNN) [1] が 1988 年に L. O. Chua らによって開発された。CNN は局所的結合の特性を持つため、LSI 化に適している。また CNN の構造は動物の網膜の構造に類似しており、様々なアプリケーションが研究されている。[2]-[5]

本研究では、ホップフィールドニューラルネットワーク (Hopfield Neural Networks: Hopfield NN) [6] の連想記憶を用いた、空間的に一様でないセルラニューラルネットワークの設計法を提案する。CNN のセル間の結合強度はテンプレートと呼ばれるパラメータによって確立されており、処理はテンプレートに

よって決まる．CNN はテンプレートの分布によって，二種類に分類することができる．もし CNN における全てのテンプレートが同じものである場合、そのシステムは“空間的に一様な CNN”となる．一方 CNN における全てのテンプレートが同じものでない場合、そのシステムは“空間的に一様でない CNN”となる．実装のコストが安いため，“空間的に一様な CNN”についての研究が盛んに行われている．しかしこの CNN に関する研究は飽和状態にある．一方，“空間的に一様でない CNN”は高度な処理を実行できるが，設計法は複雑である．しかし本手法は連想記憶を用いて適材適所にテンプレートを配置するため“空間的に一様でない CNN”を簡単に実現できる．

2 章で Hopfield NN を用いた“空間的に一様でない CNN”の設計法を提案する．3 章でコンピュータシミュレーションにより，想起パターンと既存のテンプレートが処理に大きく影響することを確認し，これまでの CNN と比べることでその効果を確認する．4 章でまとめる．

2. Hopfield NN の連想記憶を用いた空間的に一様でない CNN の設計法

これまでの“空間的に一様でない CNN”は高度な処理が可能であるが，設計法は複雑であった．しかし，本研究では Hopfield NN の連想記憶と既存のテンプレートを用いるため簡単である．以下に設計方法を示す．

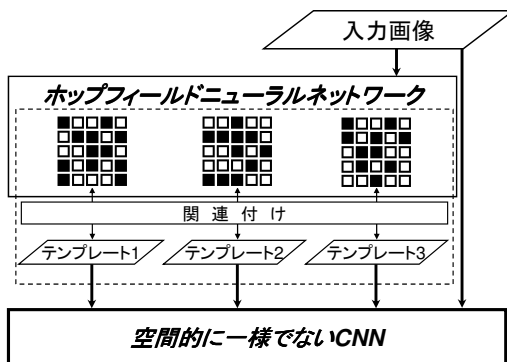


図 1 Hopfield NN を用いた空間的に一様でない CNN の設計法．

図 1 は設計の概要である．最初に，いくつかの任意の二次元パターンを準備し，それぞれのパターンに既存のテンプレート A_i, B_i, I_i を図 2 のように関連付けておく．

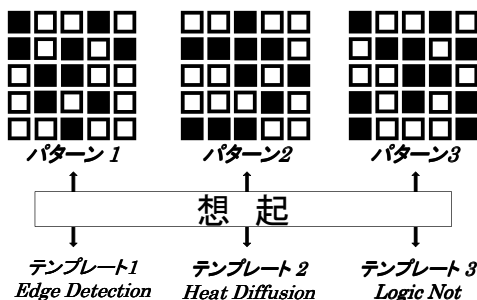


図 2 任意のパターンとテンプレートの関連付け．

次にそれらのパターンを Hopfield NN に記憶させ，結合強度

w を生成する．本研究では 2 値画像を用いる．

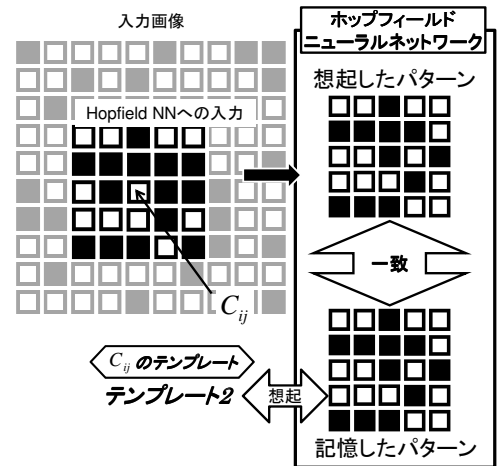


図 3 想起したパターンに基づくテンプレート配置．

図 3 は想起されたパターンによる，テンプレートの決定の過程である．入力画像における，あるピクセル C_{ij} とその 2 近傍の画素を，3 つのパターンを記憶させた Hopfield NN に入力する．Hopfield NN は記憶したパターンまたはその反転パターンの中で，入力されたパターンに最も近いパターンを想起する．これによって C_{ij} には想起したパターンに基づいてテンプレートが与えられる．例として図 3 では， C_{ij} とその 2 近傍の画素は，パターン 2 を想起している．よって C_{ij} にはパターン 2 に関連付けられたテンプレート 2 が与えられる．

C_{ij} にテンプレート 2 が与えられるので，状態方程式は以下のように変化する．

状態方程式:

$$\frac{dv_{xij}}{dt} = -v_{xij} + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} A_{2(i,j;k,l)} v_{ykl}(t) + \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} B_{2(i,j;k,l)} v_{ukl}(t) + I_2. \quad (1)$$

Hopfield NN では想起とテンプレート配置のプロセスを全てのセルにおいて実行する．以上の過程によって“空間的に一様でない CNN”は設計される．このシステムにおいて CNN は，セル毎に与えられたテンプレートを用いて処理を行うことになる．

3. シミュレーション結果

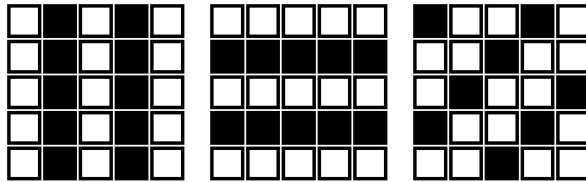
提案手法を用いてシミュレーションを行う．テンプレートは全て文献 [7] から引用する．

3.1 テクスチャー分割

この節では，3 種類のテクスチャーに対してそれぞれ異なるテンプレートを与え，テクスチャーを分割する．

パターン 1 : “White Filler” テンプレート:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = 0, I = -4. \quad (2)$$



(a) パターン 1. (b) パターン 2. (c) パターン 3.

図 4 記憶させるパターン.

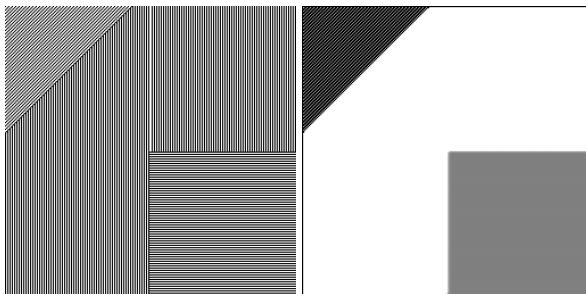
パターン 2 : “Heat Diffusion” テンプレート:

$$A = \begin{bmatrix} 0.1 & 0.15 & 0.1 \\ 0.15 & 0 & 0.15 \\ 0.1 & 0.15 & 0.1 \end{bmatrix}, B = 0, I = -1. \quad (3)$$

パターン 3 : “Logic Not” テンプレート:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I = 0. \quad (4)$$

図 4 のようなパターンを準備し、それぞれに式 (2)–(4) のテンプレートを関連付けておく．3 つのパターンは縦、横、斜め方向の直線に対応している．Hopfield NN がパターン 1, 2 または 3 を想起した時、そのセルには “White Filler” テンプレート, “Heat Diffusion” テンプレート, “Logic NOT” テンプレートがそれぞれ与えられる．このシミュレーションでは、記憶したパターンを想起し、それぞれのセルにテンプレートが与えられているかの確認を行う．



(a) (b)

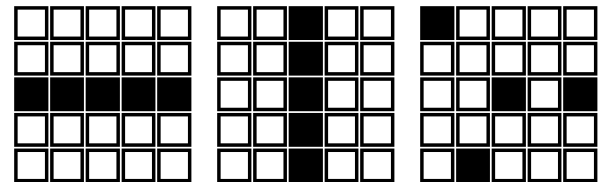
図 5 提案手法を用いたテクスチャー分割のシミュレーション結果.
(a) 入力画像 (256 × 256). (b) 提案手法を用いた出力画像.

図 5 (a) が縦、横、斜め方向の直線で構成されたテクスチャー画像 (入力画像) である．図 5 (b) がテクスチャー分割のシミュレーション結果である．縦方向の直線の領域には “White Filler” テンプレートが与えられ、それによってその領域は白く塗りつぶされている．横方向の直線の領域には “Heat Diffusion” テンプレートが与えられ、その領域は様な濃度となっている．さらに斜め方向の直線の領域には “Logic NOT” テンプレートが与えられたことで領域の白と黒が反転している．以上によってテクスチャー分割を実現した．シミュレーション結果から、それぞれのテクスチャー領域がそれぞれのパターンを想起し、それぞれのテンプレートで処理を行っていることがわかる．さ

らにこの結果から CNN はテンプレートによる処理に依存しているため、テクスチャーの境界付近セルにおいても、それぞれの処理を維持していることがわかる．つまり、一度の処理の中で 3 つのテンプレートが共存しているということになる．

3.2 エッジ検出とノイズ除去

この節では、エッジ検出とノイズ除去という相反する処理を同時に実行する．



(a) パターン 1. (b) パターン 2. (c) パターン 3.

図 6 記憶させるパターン.

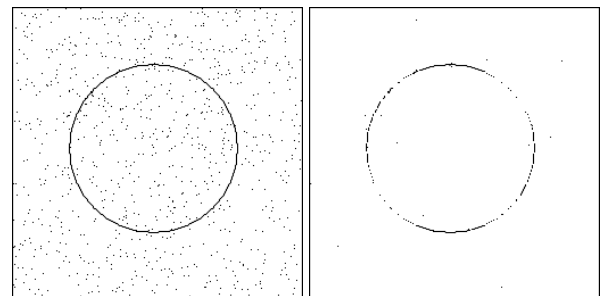
パターン 1 及び 2 : “Logic AND” テンプレート:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I = -1. \quad (5)$$

パターン 3 : “Small Object Remover” テンプレート:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, B = 0, I = 0. \quad (6)$$

図 6 のようなパターンを準備し、それぞれに式 (5), (6) のテンプレートを関連付けておく．3 つのパターンは縦、横方向の直線とノイズを模したパターンである．Hopfield NN がパターン 1 または 2 を想起した時、そのセルには “Logic AND” テンプレートが与えられ、ノイズを模したパターンを想起したセルには “Small Object Remover” テンプレートが与えられる．



(a) (b)

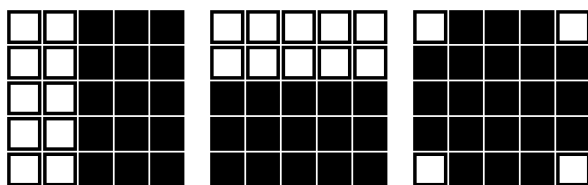
図 7 提案手法を用いたエッジ検出とノイズ除去のシミュレーション結果.
(a) 入力画像. (b) 提案手法を用いた出力画像.

図 7 (a) がノイズを含んだ円の画像である．図 7 (b) は直線検出とノイズ除去を実行した結果の画像である．円に含まれるパターン 1 または 2 を想起したセルに、“Logic AND” テンプレートが与えられている．さらにここで、斜め方向の要素を含んでいる部分に関して “Logic AND” テンプレートが与えられ、エッジの検出を行っている．これは想起のあいまいさ

が関係している．連想記憶は記憶させたパターンに最も近いパターンを想起するため，いくつかの入力パターンは円の斜め方向の要素を含んでいるとしても，縦または横方向のパターンを想起している．つまり，記憶したパターンと入力したパターンが完全一致でなくても，いくつかのセルに“Logic AND”テンプレートが与えられた．また，ノイズを持つセルのほとんどに“Small Object Remover”テンプレートが与えられ，ノイズ除去が実行された．通常このような画像に対してノイズ除去を実行すると，円のエッジも消えてしまう．しかし，円のエッジに“Logic AND”テンプレートを与えることで，的確にノイズ除去を実行することができた．この結果から，エッジ検出とノイズ除去という相反する処理でも平行して行えることが確認できた．

3.3 反転したエッジの検出

この節では，複数の CNN を用いなければ得られない出力画像を一度の処理で得る．



(a) パターン 1. (b) パターン 2. (c) パターン 3.

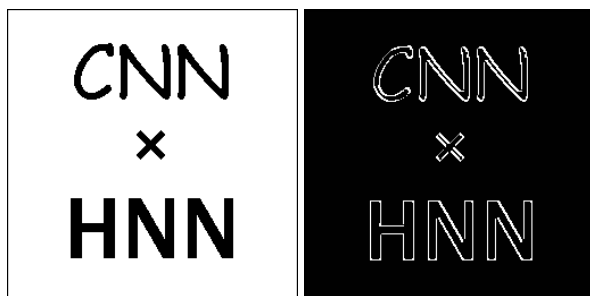
図 8 記憶させるパターン.

パターン 1 及び 2: “Logic NOT” テンプレート (4)

パターン 3: “Black Propagation” テンプレート:

$$A = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix}, B = 0, I = 3.75. \quad (7)$$

図 8 のようなパターンを準備し，それぞれに式 (4)，(7) のようなテンプレートを関連付けておく．3 つのパターンは縦，横方向のエッジ付近のパターンと，一様な領域のパターンである．Hopfield NN がパターン 1 または 2 を想起した時，そのセルには“Logic NOT”テンプレートが与えられ，パターン 3 を想起したセルには“Black Propagation”テンプレートが与えられる．



(a)

(b)

図 9 提案手法を用いた反転したエッジの検出シミュレーション結果．

(a) 入力画像. (b) 提案手法を用いた出力画像.

このシミュレーションでは，入力画像に図 9 (a) のような文字画像を入力する．図 9 (b) がシミュレーション結果である．文字と背景の境界付近においてパターン 1 及び 2 を想起している．Hopfield NN は記憶させたパターンが反転したパターンを出力する場合もあるため，文字の上下左右のエッジ付近においてパターン 1 または 2，またはその反転パターンを想起している．また，3.2 でも示したように，本設計方法では Hopfield NN の連想記憶の機能を用いているため，斜め方向成分を含んでいる入力パターンにおいても，パターン 1 または 2，またはその反転パターンを想起するケースが多い．これらによって，文字のエッジのほぼ全てのセルに“Logic NOT”テンプレートを与えることができた．また，一様な領域には“Black Propagation”テンプレートが与えられている．これによって“Logic NOT”テンプレートを与えられたセル以外は黒く塗りつぶされた．通常の CNN を用いてこのような出力画像を得ようとする場合，エッジ検出を実行した画像を別の CNN を用いて反転する必要がある．本手法では，この処理を 1 つの CNN で実行できるため効果的であるといえる．

4. ま と め

本研究では，ホップフィールドニューラルネットワークの連想記憶を用いて空間的に一様でないセルラニューラルネットワークの設計法を提案した．提案手法はこれまでの“空間的に一様でない CNN”の設計法と比べて簡単である．また，Hopfield NN の連想記憶の機能を用いて複数のテンプレートを適材適所に与えることができる．その結果，これまでの CNN では複数回処理を行わなければ得られなかった画像を一度の処理で得られるなどの効果を確認した．本手法はテンプレートと想起パターンによって処理が変わるため，様々な画像処理に適すると考えられる．

文 献

- [1] L. O. Chua and L. Yang, “Cellular Neural Networks: Theory,” IEEE Trans. Circuits Syst., vol. 32, pp. 1257-1272, Oct. 1988.
- [2] F. Dirk and T. Ronald, “Coding of Binary Image Data using Cellular Neural Networks and Iterative Annealing,” Proc. of ECCTD’03, vol. 1, pp. 229-232, Sep. 2003.
- [3] M. Namba and Z. Zhang, “Cellular Neural Network for Associative Memory and Its Application to Braille Image Recognition,” Proc. of IJCNN’06, pp. 4716-4721, Jul. 2006.
- [4] H. Koepl and L.O. Chua, “An Adaptive Cellular Nonlinear Network and its Application,” Proc. of NOLTA’07, pp. 15-18, Sep. 2007.
- [5] K. Sumitomo and A. Ushida, “A Design Method of Cellular Neural Networks using Fuzzy Inference,” IEICE Technical Report, NLP, vol. 98, no. 443, pp. 39-46, Dec. 1998.
- [6] J. J. Hopfield, “Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons,” Proc. Natl. Acad. Sci. USA, vol. 81, pp. 3088-3092, 1984.
- [7] Cellular Sensory Wave Computers Laboratory Computer and Automation Research Institute Hungarian Academy of Sciences, “Cellular wave Computing Library (Template, Algorithms, and Programs) Version 2.1”