

Small-World Cellular Neural Networks における多重化による フォルト・トレランス実装の提案と評価

松本 勝慶† 上原 稔† 森 秀樹†

† 東洋大学大学院工学研究科情報システム専攻

Cellular Neural Networks(CNN)において、Small-World ネットワークのネットワーク構造を導入することにより、CNN をより効率的に処理が行える Small-World Cellular Neural Networks(SWCNN) が提案されている。CNN を Small-World 化することにより、CNN の特定のタスクを効率的に行うことが可能となる。しかし、この提案されている SWCNN は、故障やエラーが発生した場合に対処が考慮されていない。そこで、本研究では、多重化技法を用いることにより、SWCNN の耐故障性を高める方法を提案する。

Evaluation and Proposal implementation of Fault Tolerance techniques for Small-World Cellular Neural Networks

Katsuyoshi Matsumoto †, Minoru Uehara †, Hideki Mori †

† Department of Open Information Systems Graduate School of Engineering Toyo University

In this paper, we propose Small-World Cellular Neural Networks (SWCNNs) featuring multiplex fault tolerant techniques. SWCNN is a Cellular Neural Networks in which cells are connected by small-world connections. SWCNN can easily process many multimedia applications such as image processing. However, SWCNN needs to be fault tolerant because it displays higher error propagations than traditional CNNs. We describe typical neural algorithms for image processing such as noise removal. Also, we propose a fault tolerant architecture for the CNN, using multiplexing.

1. はじめに

Cellular Neural Networks (CNN)はテンプレートを変更することにより、画像処理、パターン認識などの様々なアプリケーションが可能となる。CNN で画像処理を行う場合、Small-World ネットワークのネットワークモデルを組み込むことにより、画像処理の処理性能を向上させる Small-World CNN が提案されている[1]~[3]。しかしながら、この提案された SWCNN は故障やエラーが発生した場合、対処する方法が考慮されていない。そこで、本研究

では、多重化冗長方式を用いて、SWCNN を提案し、評価を行う。

2. Small-World Cellular Neural Networks (SWCNN)

SWCNN は、CNN に Small-World ネットワークを組み込むことにより、画像処理のタスクにおいて性能を向上させることが可能である[1]~[3]。

Small-World ネットワークは、規則的なネットワークとランダムなネットワークの間に存

在するネットワークである [1][2][4]. この Small-World ネットワークでは, ネットワーク構造を表す特徴量で C と L で示される. C はクラスタリング係数である. L は全てのノード間の最短パス長の平均である. Small-World ネットワークは, C が大きく L が小さい.

CNN は, 規則的に並べられたニューロンが局所的に結合したニューラルネットワークである. SWCNN は, CNN のネットワーク構造に Small-World ネットワークを適用させた CNN である [1][2]. 図 1 に SWCNN のネットワーク構造を示す.

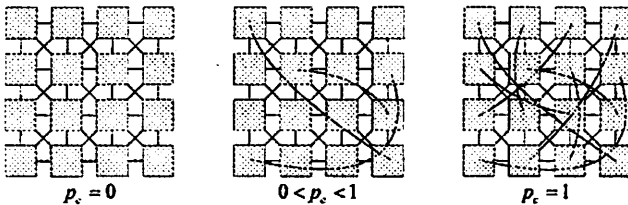


図 1 SWCNN のネットワークイメージ

このようなネットワークを形成するために, ランダムに結合する確率 p_c を導入する. ランダムに結合するノードは 1 対 1 のみに対応する. そのため, 増加するエッジはノードの総数半分である.

SWCNN の状態方程式を示す.

状態方程式

$$\begin{aligned}
 x_{ij}(t+1) = & -x_{ij}(t) + I \\
 & + \sum_{kl \in N_r(ij)} A(ij;kl) y_{kl}(t) \\
 & + \sum_{kl \in N_r(ij)} B(ij;kl) u_{kl}(t) \\
 & + M(i, j; p, q) y_{ij}(t)
 \end{aligned}$$

また, 出力方程式は以下の式である.

出力方程式

$$y_{ij}(t) = \frac{1}{2} (|x_{ij} + 1| - |x_{ij} - 1|)$$

ここで x_{ij} は状態で, y_{ij} は出力を示し, u_{ij} は入力を示す. A は出力テンプレートで B は入力テンプレートである. 出力テンプレートは, フィードバックテンプレートであり, 出力 y_{ij} に依存する. 入力テンプレートはフィードフォワードテンプレートであり, 入力 u_{ij} に依存する. $N_r(ij)$ は素子 ij の近傍を示す. kl は素子の近傍素子である. 近傍距離は r の値によって決まる. $r=1$ の場合, 3×3 となり近傍は以下の図 2 になる.

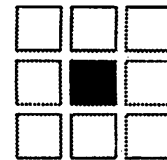


図 2 $r=1$ の時の近傍素子

I は閾値である. 出力 u_{ij} は, $-1 \sim 1$ までの値

を出力するように設計されている. 入力 u_{ij} は $-1 \sim 1$ までの値のみ入力する. CNN にスモールワールドネットワークを適用させるためにランダムに結合する確率 p_c を導入する. ランダムに結合する素子同士の結合は 1 対 1 にのみ結合する. $p_c=1$ の場合, SWCNN の増加したエッジ数は素子数の $1/2$ となる. w_c はランダムに結合する素子同士の結合係数である. $M(ij; pq)$ は素子 ij と素子 pq が結合することを示す. 確率 p_c によって $M(ij; pq)$ の結合が決定される.

3. 多重化冗長方法

3.1 二重化比較方式

二重化比較方式では、同一のモジュールを並列にし、同一データに対し、演算を同時に実行し、その出力結果を比較することにより、故障を検出する[5][6]。(図3)

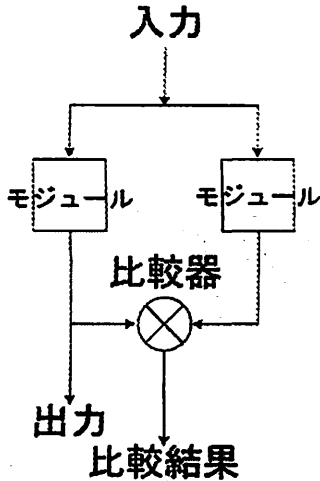


図3 二重化比較方式

二重化比較方式では、故障の検出は可能である。しかし、故障モジュールの特定は困難である。

3.2 Triple Modular Redundancy (TMR)

TMR は古くからあるエラーマスク方法である[5][6]。同一のモジュールを三重化し、その出力を多数決することにより、出力を決定する方法である。(図4)

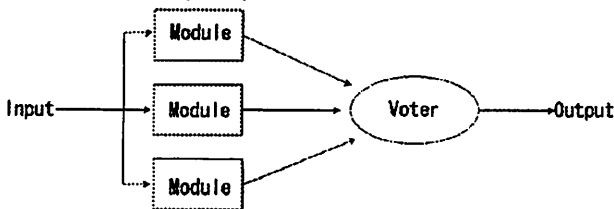


図4 TMR の概念図

4. 多重化 SWCNN

4.1 SWCNN with Duplex

本研究では、二重化比較方式を用いた方法と TMR を用いた方法を提案する。

二重化比較方式では、2つのニューロンを並

べその出力を比較し、故障を検出する。比較情報をもとに故障箇所を分離させる。分離させた箇所のニューロンは、元のデータを利用する。(図5)

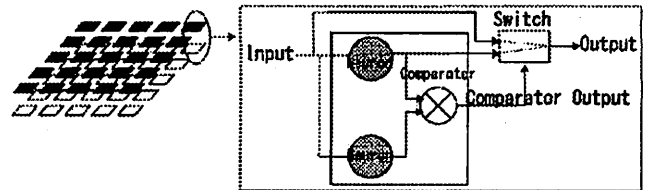


図5 SWCNN with Duplex

4.2 SWCNN with TMR

TMR 方式では、SWCNN を3つ並列にし、同一箇所のニューロンの出力を多数決することにより、SWCNN のフォルト・トレランス性を向上させる。

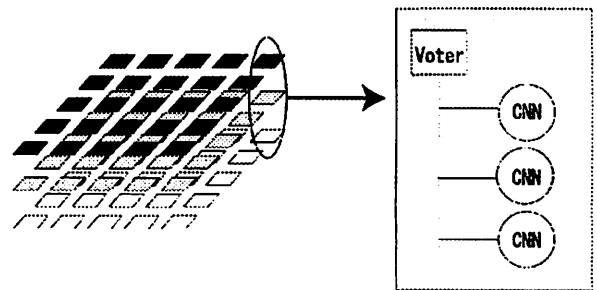


図6 SWCNN with TMR

多数決の結果が一致すれば一致した値を出力する。

5. 実験と評価

5.1 実験、評価方法

以下に本研究での実験方法と評価方法を示す。

- 故障モデル 0, 1 の固定故障
- 故障率 0~0.1
- テンプレート 画像処理
ノイズ除去
- ランダム結合確率 1
- 故障率0時の出力画像と比較

本研究では、画像処理用 SWCNN を用いて評価を行う。用いるテンプレートを以下に示す。

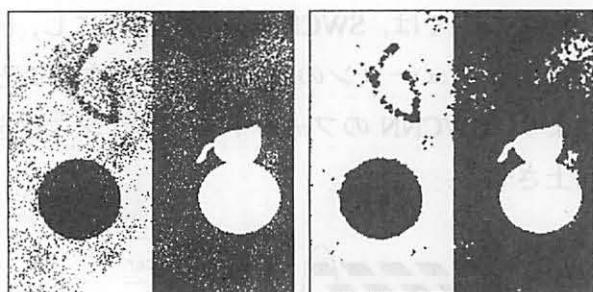
ノイズ除去

以下のテンプレートをを用いることで二値画像中のノイズを除去することが可能となる。

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$I = 0, w_c = 2.0$$

また、処理画像と故障率 0 時の出力画像を示す。

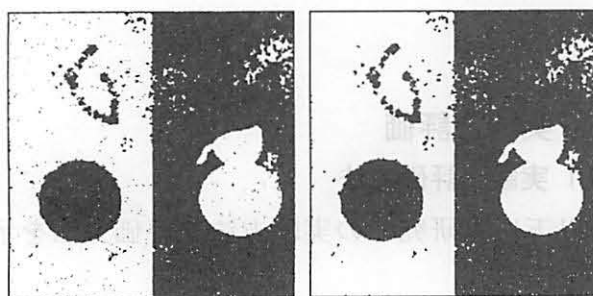


(a) (b)

図 7 原画像(a)と故障率 0 時の出力画像(b)

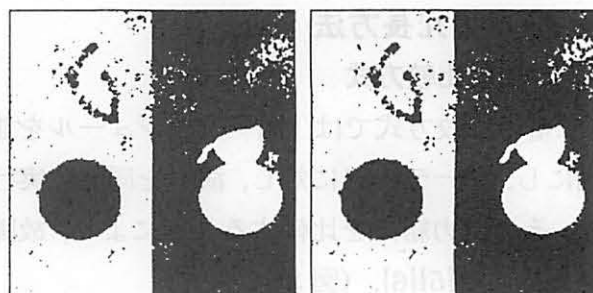
5.2 評価

以下に実行結果の出力画像を示す。



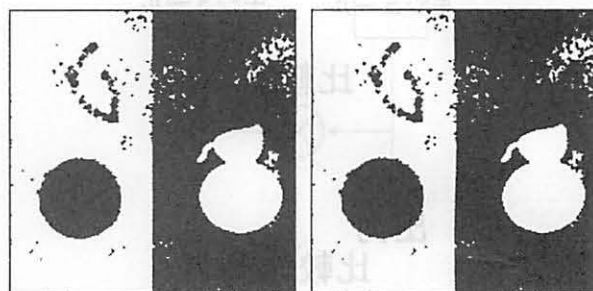
(a) (b)

図 8 SWCNN の故障率 0.01 時の 0 故障(a)と 1 故障(b)の出力画像



(a) (b)

図 9 SWCNN with Duplex の故障率 0.01 時の 0 故障(a)と 1 故障(b)の出力画像



(a) (b)

図 10 SWCNN with TMR の故障率 0.01 時の 0 故障(a)と 1 故障(b)の出力画像

故障率 0.01 時の出力画像では、フォルト・トレランス実装を行っていない SWCNN が故障によるエラー出力が 0 故障, 1 故障両方の場合とも目立っている。フォルト・トレランス実装を行った SWCNN with Duplex, SWCNN with TMR どちらも故障を抑えることができた。しかし、SWCNN with Duplex では、故障箇所は検出することが可能だが、故障箇所に元のデータを利用するため、わずかであるが不一致となる。

図 11 に実装の違いによる出力画像の出力画素の不一致度を示す。

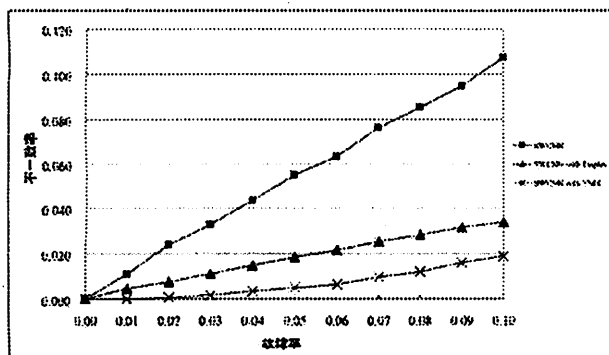


図 11 不一致率 0 固定故障

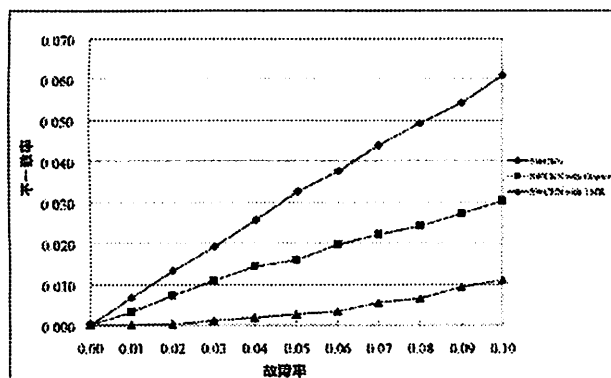


図 12 不一致率 1 固定故障

図 11, 12 のグラフでは, 故障率 0 時の出力画像と故障率を変化させた出力画像を画素ごとと比較したグラフである. 不一致率は以下の式に示す.

$$\text{不一致率} = \frac{\text{不一致画素数}}{\text{すべての画素数}}$$

故障率を変化させることにより, 不一致率が増加する. また, 0 故障と 1 故障での不一致率の違いは, 0 故障での故障箇所出力が灰色となり, 1 故障での故障箇所出力が白色となるため, 不一致率は 0 故障のほうが大きくなる.

このグラフを見ても, フォルト・トレランス実装を行っていない SWCNN では不一致率が最も高くなっている. フォルト・トレランス実装を行うことで, SWCNN with Duplex では半分, SWCNN with TMR では, 1/5 に不一致率を抑えることができた.

6. まとめ

本研究では, SWCNN において多重化を用い

てフォルト・トレランス実装し評価を行った. 多重化を行うことにより, SWCNN のフォルト・トレランス性は向上した. しかし, TMR では, モジュールが 3 倍必要である. また, Duplex では, 故障箇所を検出し, 分離させるのみであり故障箇所を復旧させるまでには, 至っていない.

今後の課題としては, Duplex 方式で故障モジュールの特定を行い, 故障モジュールの復旧が可能となる方法が求められる.

参考文献

- [1] Kazuya Turuta, Zonghuan Yan, Yoshifumi Nishio, Akio Ushida, "Small-World Cellular Neural Networks for Image Processing Applications", ECCTD'03-European Conference on Circuit Theory and Design, vol. 1, pp225-228
- [2] Kazuya Turuta, Zonghuan Yang, Yoshifumi Nishio, Akio Ushida, "On Two Types of Network Topologies of Small-World Cellular Neural Networks", RISP International Workshop on Nonlinear Circuit and Signal Processing (NCSP'04), pp. 113-116, Mar. 2004
- [3] L. O. Chua, L. Yang, "Cellular Neural Networks: Theory", IEEE Transactions Circuit and Systems vol. 35, 10, pp1257-1272, 1988
- [4] Duncan J. Watts "Small Worlds", Princeton University Press, 1999
- [5] Dhira K. Pradhan, "Fault-Tolerant Computer System Design", Prentice Hall PTR, New Jersey, 1996
- [6] 向殿政男, 秋田雄志, 奥村幸一, 尾崎俊治, 菊野亨, 古賀義亮, 南谷崇, 藤原秀雄, 窪原弘一, 船津重弘, 山田茂, "フォルト・トレラント・コンピューティング", 丸善, 1989