

Title	階層型ニューラルネットワークのニューロン故障補償手法の実装
Author(s)	菅原, 英子; 堀口, 進
Citation	電子情報通信学会論文誌 C, J85-C(9): 861-864
Issue Date	2002-09-01
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/4723
Rights	Copyright (C)2002 IEICE. 菅原 英子, 堀口 進, 電子情報通信学会論文誌 C, J85-C(9), 2002, 861-864. http://www.ieice.org/jpn/trans_online/
Description	

階層型ニューラルネットワークのニューロン故障補償手法の実装

菅原 英子^{†a)}(学生員) 堀口 進[†](正員)

Hardware Implementation of Hierarchical Neural Networks with Neuron Defect Compensations

Eiko SUGAWARA^{†a)}, Student Member and

Susumu HORIGUCHI[†], Regular Member

[†] 北陸先端科学技術大学院大学, 石川県

Japan Advanced Institute of Science and Technology, Asahidai 1-1, Tatsunokuchi-machi, Nomi-gun, Ishikawa-ken, 923-1292 Japan

a) E-mail: esugawa@jaist.ac.jp

あらまし 近年, VLSI 技術の発展により, ニューラルネットワークのハードウェア実装に関する研究が行われている. ニューラルネットワークのハードウェア実装には, ハードウェア故障への対処法が不可欠である. 本論文では, 予備ニューロンを用いたハードウェア故障補償を行う機能シフト法を提案し, FPGA による実装を行い, 回路評価について議論した. その結果, 故障補償のための回路量は比較的少なく, 高速なハードウェア故障の回避が可能であることがわかった.

キーワード 階層型ニューラルネットワーク, 故障補償, ハードウェア実装, FPGA

1. まえがき

ニューラルネットワーク (Neural Network: 以下, NN) はパターン認識やロボット等の制御などの分野で幅広く応用されている. また, 近年の VLSI 技術の発展に伴い, NN そのものをハードウェア上に実装する研究が行われている.

NN のハードウェア実装においては回路規模とハードウェア故障を考慮する必要があり, 様々な回路規模削減手法が提案されている [1], [2]. 一方, ハードウェア故障は避けることができないため, 効率の良い故障補償手法に関する研究の重要性が高まっている.

安永ら [3] は学習を繰り返すことで故障箇所を切り離すことができることを示した. また, Khunasaraphan ら [4] は再学習によらない重み更新手法を提案した. しかし, これらの重み更新による故障回避には長時間を要することや, 特別な回路の付加が必要であるといった問題が指摘されている. そこで, 本論文では予備ニューロンを用いて故障箇所を切り換えるハードウェア故障補償である機能シフト法を提案する. ここでは Skubiszewski [5] によるバイナリ NN をもとに, 乗算

器を用いない階層型 NN の回路設計を行い, 機能シフトによる故障補償の回路規模について評価を行う. 更に FPGA 上に階層型 NN を実装し, 故障補償回路の動作検証を行う. 対象とするネットワークは学習機能をもたないフィードフォワード型 3 層階層型 NN であり, 計算機上で学習された重みを与える.

2. ニューロン故障補償を行う階層型 NN 回路

階層型 NN では, 各ニューロンは隣接する層のすべてのニューロンと完全結合される. そのため, ネットワークの大規模化に伴いニューロン間結線は大幅に増加し, 大規模 NN のハードウェア実装やハードウェア故障補償は非常に困難になる.

そこで, 入出力切換をパイプライン的に行い, ニューロン間結線を少なくしハードウェア故障補償を行う階層型 NN を考える. 図 1 に示すように中間層, 出力層には予備ニューロンを配置し, 各層間に入力選択のためのセクタ回路を配置する. セクタは下位層のニューロンの出力を一つずつ選択し, 上位層の全ニューロンに同時に入力を与える. 各ニューロンは重みを格

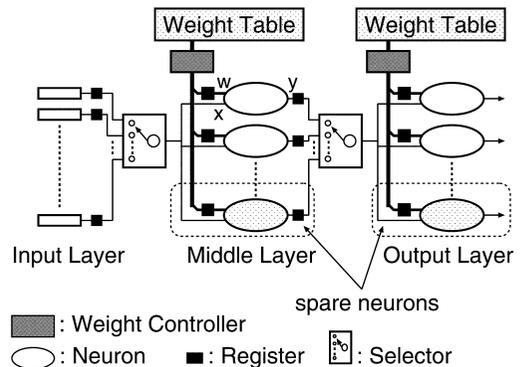


図 1 階層型 NN の回路構成
Fig. 1 Concept of a hierarchical neural network.

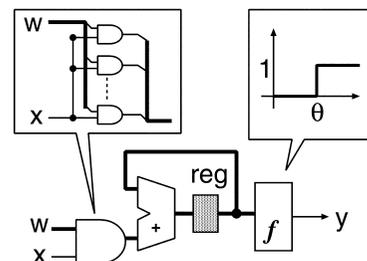


図 2 ニューロンの回路構成
Fig. 2 Configuration of circuit of neuron.

納するためのレジスタをもち、重み獲得回路（図1の Weight Controller）を通して Weight Table から対応する重みを得る。重み獲得回路は各ニューロンの状態をもとに対応する重みを決定する回路であり、故障ニューロンの重みをシフトさせる機能を有している。Weight Table はあらかじめ計算機上で学習された重みを保持するテーブルである。これらの故障補償手法については次章で詳しく述べる。

図2は階層型 NN 回路内で用いるニューロンの回路構成を示す。活性化関数として2値のステップ関数を使用しており、入力 x とそれに対応する重み w の乗算を行う回路として論理積回路を用いることでニューロン回路を簡略化している。

図1の $NN(l, m, n)$ は入力層に l 個、中間層に m 個、出力層に n 個のニューロンを有する階層型 NN である。入力層ニューロンの l 個の出力はセクタを通して一つずつ中間層の全ニューロンに同時に入力される。中間層では m 個のニューロンが並列動作し、その後、中間層ニューロンの m 個の出力が出力層に送られ、 n 個のニューロンが並列に処理する。したがって、一つの入力データセットに対し、出力が得られるまでにニューロンの積和演算に要する時間 τ_p は、2変数の乗算1回当たりの時間を t_m 、2変数の加算1回当たりの時間を t_a とすると

$$\tau_p = (l + m)(t_m + t_a) \quad (1)$$

となる。これに対し、完全結合型 NN 回路においてニューロンの積和演算に要する時間 τ_c は

$$\tau_c = 2t_m + t_a(\log l + \log m) \quad (2)$$

となる。ここで、それぞれの回路規模を考える。完全結合型 NN 回路では k 入力ニューロン1個当り k 個の2入力1出力乗算回路と $(k-1)$ 個の2入力1出力加算回路を必要とし、各層間の結線数も(入力数) \times (ニューロン数)と膨大になる。これはハードウェア実装が不可能なほどの回路規模である。これに対し、セクタ型 NN 回路では、入力数によらず1ニューロン当り乗算回路と加算回路を一つずつしか必要としない上に、各層間の結線数も入力数分ですむという利点がある。

3. 機能シフト法によるニューロン故障補償

3.1 機能シフト法

機能シフト法は故障ニューロンを取り除き、必要とする階層型 NN を再構成する手法である。論理的な NN の構成 $NN(l, m, n)$ に対し、物理的な NN の構成

を $NN(L, M, N)$ とおく。物理ニューロンの中から中間層に s 個、出力層に t 個の予備ニューロンを割り当て、 $NN(l, m, n) = NN(L, M-s, N-t)$ とする。故障ニューロンがあると、故障ニューロンよりも下位のニューロンへ順次機能がシフトされていく。

一般に NN の故障として、ニューロン、重み、リンクの故障の3種類が挙げられる。機能シフト法は予備ニューロンを用いて故障ニューロンを切り離す手法であり、重みやリンクをニューロンの一部とみなし、これらすべての故障をニューロン故障として扱う。

3.2 機能シフト法による重み獲得

予備ニューロンを含め、中間層、出力層に存在するすべてのニューロンには層ごとに物理 ID (P_{id}) と論理 ID (L_{id}) が与えられる。 P_{id} は故障ニューロンも含めた通し番号で、 L_{id} は故障ニューロンを除いたニューロンの通し番号である。あるニューロンが故障しているか否かは故障検出回路の状態フラグ ($state$) として各ニューロンが保持する。各ニューロンの状態フラグを用いて計算された故障数 ($brkn$) も各ニューロンが保持する。ここで、故障数はあるニューロンにおいてそれより上位のニューロンにいくつ故障が発生しているかを示す値である。機能シフト法における重み獲得手順を以下に示す。

(1) 各ニューロンの論理 ID の初期化

$$L_{id}(i) = P_{id}(i).$$

(2) 各ニューロンの状態チェック

$$state(i) = 0 \text{ (正常)}, 1 \text{ (故障)}.$$

(3) 故障数のカウント

$$brkn(i) = \sum_{i=0}^P state(i),$$

$$P = M - 1 \text{ (中間層)}, N - 1 \text{ (出力層)}.$$

(4) 各ニューロンの論理 ID の更新

$$L_{id}(i) = P_{id}(i) - brkn(i).$$

(5) 更新された論理 ID に対応する重みを Weight Table から獲得する。故障ニューロンには0を与える。

手順(1)~(4)は回路の初期化時に行われる。手順(5)は各層での演算実行時に行われる。

図3に重みの獲得例を示す。これは中間層若しくは出力層において実行に必要なニューロン数が3個、予備ニューロンとして2個のニューロンが配置されており、 $P_{id} = 1$ のニューロンが故障していると仮定した例である。Weight Table 内の $ID = 1$ に割り当てられる重みは $L_{id} = 1 (P_{id} = 2)$ のニューロンに、 $ID = 2$ に割り当てられる重みは $L_{id} = 2 (P_{id} = 3)$ のニューロ

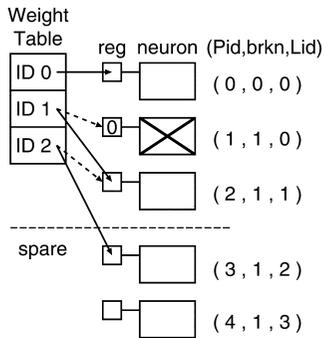


図3 機能シフト法による重みの獲得例
Fig.3 Example of functional shifting.

にそれぞれ与えられる． $P_{id} = 1$ の故障ニューロンには重みとして0が与えられる．

4. FPGAによるニューロン故障補償を行う階層型NNのハードウェア実装

4.1 実装環境

ニューロン故障補償を行うNN回路をFPGA上に実装した．実装環境はALTERA社の二つのFPGAで，入出力用としてEPF10K10QC208-4(10Kgate)を使用し，設計した階層型NN回路の実装にはEPF10K130VGC599-3(130Kgate)を使用した．

回路はハードウェア記述言語であるVHDLで記述し，ALTERA社のMAX+Plus IIを用いて回路設計，シミュレーションを行いEPF10K130VGC599-3に実装し，回路規模に関して検討した．

4.2 回路構成の比較

ニューロン故障補償有/無の階層型NNの回路構成を表1に示す．なお，本論文では，故障検出回路そのものは実装していない．ニューロン故障補償を行うNN回路は，故障数カウント回路，重み獲得回路，及びニューロンの状態などを保持するためのレジスタが追加されている．また，中間層・出力層間に配置するセレクトに，中間層に存在する故障ニューロンの出力を削除する機能を追加している．入力層・中間層間に配置するセレクトは故障補償を行わないNN回路のものと同様のものである．

ニューロンの回路構成は故障補償有/無にかかわらず等しいが，実行に使用できるニューロン数が異なる．故障補償を行わないNN回路は最大で $NN(L, M, N)$ の実行が可能であるのに対し，故障補償を行うNN回路は縮退型であり，中間層，出力層で使用できるニューロン数が予備ニューロンの分だけ少なくなる．中間層

表1 故障補償有/無の回路構成

Table 1 Circuit difference of defect compensation.

故障補償無 NN 回路	故障補償有 NN 回路
ニューロン	ニューロン
入力層/中間層セレクト	入力層/中間層セレクト
中間層/出力層セレクト	中間層/出力層セレクト
Weight Table	Weight Table
—	故障数カウント回路
—	重み獲得回路
レジスタ	レジスタ

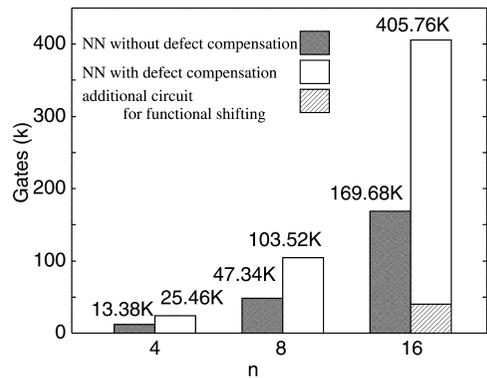


図4 故障補償有/無の回路規模の比較
Fig.4 Comparison of the number of gates.

に s 個，出力層に t 個の予備ニューロンを配置したとすると， $NN(L, M - s, N - t)$ が実行可能である．

4.3 回路規模の評価

図4に重みを8ビットとした故障補償有/無の階層型NN全体の回路規模を示す．縦軸は回路規模(ゲート数)を表し，横軸の n は各層に配置したニューロン数を表す．ここでは， $n = L = M = N$ であり，故障補償有NNでは n に予備ニューロンも含まれる．故障補償有NNの回路規模は故障補償無NNの約2倍になった．図4における $n = 16$ のグラフの斜線部は，レジスタを除く機能シフト法のために追加した回路量を示す．この回路量は全体の約10%程度で，故障数カウント回路や重み獲得回路の占める割合は比較的少ない．なお， $n = 4, 8$ のときの追加回路量はほとんどがレジスタである．

機能シフト法ではニューロンの物理IDと論理ID，故障数，状態フラグといったデータを使用する．これらのデータは2進数で表され，ニューロンごとに各データがそのままレジスタに格納されている．ニューロン数が増えるとレジスタ数及び各レジスタのビット長が

増加するため、 $O(n \log n)$ ビットのレジスタ容量が必要である。

5. む す び

本論文では、ニューロン故障補償を実現するための機能シフト法を提案し、回路設計とFPGA実装を行い、回路規模の評価と動作検証を行った。

セレクタを配置したNNの回路構成は各層間を完全結合したNNと比較して処理速度は劣るものの、回路量が非常に少なく、ハードウェア実装可能であり、故障補償容易性という観点からも優れている。レジスタ数増加の問題に対しては、各データを符号化することによってレジスタ数を削減したり、レジスタを使用せずにデータをメモリに書き出すといった方法で機能シフト法による故障補償回路をより小規模回路で実現可能である。

今後は機能シフト法のデータ保持方法やデータ圧縮方法、予備ニューロンの配置方法を検討し、より効率

の良い故障補償を実現することが課題である。

謝辞 本研究の一部は日本学術振興会 科学研究費助成を用いて行われた。関係各位に深謝する。

文 献

- [1] 肥川宏臣, “ハードウェア化に適した学習機能付き3値多層ニューラルネットワーク,” 信学論 (D-II), vol.J81-D-II, no.12, pp.2811–2818, Dec. 1998.
- [2] 川島 毅, 石黒章夫, 大熊 繁, “小規模回路で実現可能なニューラルネットワークのハードウェア化手法,” 信学技報, NC99-90, Feb. 2000.
- [3] 安永守利, 浅井光男, 柴田克成, 山田 稔, “ニューラルネットワーク集積回路の自律的な欠陥救済能力,” 信学論 (D-I), vol.J75-D-I, no.11, pp.1099–1108, Nov. 1992.
- [4] C. Khunasaraphan, K. Vanapipat, and C. Lursinsap, “Weight shifting techniques for self-recovery neural networks,” IEEE Trans. Neural Networks, vol.5, no.4, pp.651–658, July 1994.
- [5] M. Skubiszewski, “A hardware emulator for binary neural networks,” International Neural Network Conference, vol.2, pp.555–558, Paris, July 1990.

(平成13年12月4日受付, 14年3月22日再受付)