

分散機械学習 MicroDeep の エネルギーハーベスト実装と実証実験

Gereltod Sengun¹ 山口 弘純² 東野 輝夫² 安本 慶一³ 田上 敦士⁴

概要: Beyond 5G 時代の将来 IoT 環境では、超多数の IoT 機器の偏在による実世界のセンシング・解析と理解、ならびにフィードバックのループであるサイバーフィジカルシステムの実現が期待されている。しかし超多数の IoT 機器からのデータ集約に伴うクラウド側データ処理やセンサデータ伝送電力の増大、プライバシーの社会問題化が懸念されている。これに対し、我々の研究グループでは、これまでクラウド側で実行されていた機械学習機能をエッジ側の端末連携で自律分散的に実現する機構である MicroDeep の開発を進めている。MicroDeep では、無線センサネットワークにおいて、個々のセンサノードが有する限られたメモリや計算機能を連携させて機械学習処理を行うとともに、それらを環境発電により動作させることで、機械学習処理をカーボンフリーで実行する概念を提唱している。本研究ではこの概念を実証する試みを紹介する。まず、太陽光発電パネルと省電力マイコン、省電力人感センサ、ならびに蓄電池を有する環境発電駆動型のセンサノード「MD-Board」を開発し、それらの連携動作を実現する MicroDeep 実行基盤を開発した。この MicroDeep 実行基盤上に軽量 CNN を分散実装し、各センサノードの人感センサの検出値に基づくセンサ領域内の人物の転倒検知機能を実現した。36 台の MD-Board からなる MicroDeep 実行基盤において人物転倒検知実験を実施した結果、太陽光が得られる明るい環境下において環境発電のみで転倒検知判定が実行可能であることを確認した。また、各センサノードの電力消費量を測定した結果、消費電力は最大で 88.4mW であり、明るい屋内における MD-Board の太陽光発電パネルの発電量とほぼ同等であることがわかった。現在、MicroDeep 実行基盤を奈良先端科学技術大学院大学が有するスマートホーム環境に設置し、連続稼働実験を実施している。

キーワード: エッジコンピューティング, センサネットワーク, 深層学習, 分散化

1. はじめに

世界のデジタルデータは飛躍的に増大しており、特に IoT 機器からのデータ増大が予想されている。スマートシティをはじめとする持続可能な環境の実現には IoT 機器は不可欠な要素であり、増大するモバイル機器や IoT 機器からのデータをいかに効率よく処理するかが IT インフラの永続的な発展のキーポイントとなる。特に今後は IT インフラの中核として、サイバーフィジカルシステムとよばれる基盤システムが一般的になると考えられる。サイバーフィジカルシステムは、実世界のあらゆる場所やモノにセンサーを浸透させ、実世界情報を活用することで、交通、

物流、エネルギーなどの高度な制御を実現するシステムであり、多数の機器やセンサーからのデータをビッグデータとしてクラウドストレージに蓄積し、計算時間と計算資源を大量に投入して緻密な AI 分析を行い、知識やパターンを抽出する集中型クラウド処理を前提としている。

しかし、IT 機器や AI の普及による消費電力増大は全世界的な課題である。特に近年では AI の訓練や実行による CO₂ 増加問題が注目を集めており、例えば 2017 年の Alpha Go Zero の学習にかかる計算回数は 2012 年の 30 万倍であったという報告もある [1]。また、世界のデータセンター消費電力は全電力の約 0.6% と依然として無視できず [2]、センシングや AI 解析の電力抑制につながるデータ量抑制に向けた取組みが将来のカーボンニュートラル社会には必須といえる。

この課題に対しては、すべてのセンシングデータをクラウド側に運搬するのではなく、可能な限りデータ発生源であるエッジ側で処理を行うエッジコンピューティングのア

¹ 大阪大学基礎工学部情報科学科
Osaka University, Japan

² 大阪大学大学院情報科学研究科
Osaka University, Japan

³ 奈良先端科学技術大学院大学
Nara Institute of Science and Technology, Japan

⁴ 株式会社 KDDI 総合研究所
KDDI Research, Inc., Japan

アプローチが普及してきており、データ処理遅延、データ伝送負荷やコネクション数の抑制効果が期待される。一方、エッジコンピューティングにおいても機械学習に要する計算処理資源は同じであり、計算にかかる消費電力の観点からはクラウドコンピューティングとの本質的な差異はない。

これに対し我々の研究グループでは、エッジにおける機械学習機能を、複数の端末連携で自律分散的に実現する機構である MicroDeep の開発を進めている [3,4]。MicroDeep では、無線センサネットワークにおいて、個々のセンサノードが有する限られたメモリや計算機能を連携させて機械学習処理を行う。機械学習処理を複数ノードにできるだけ公平に分散させることで、個々のセンサノードのメモリ制約や処理資源制約のもとでの機械学習処理を実現し、さらに、それらを環境発電により動作させることで、カーボンフリーで学習や判定を実行する概念を提唱している。

本研究ではこの概念に基づき実装した MicroDeep 実行基盤を用いた実証実験について報告する。まず、太陽光発電パネルと省電力マイコン、省電力人感センサ、ならびに蓄電池を有する環境発電駆動型のセンサノードである MD-Board を開発し、それらを複数台連携可能な MicroDeep 実行基盤を開発した。この MicroDeep 実行基盤上に軽量 CNN を分散実装し、各センサノードの人感センサの検出値に基づくセンサ領域内の人物の転倒検知機能を実現した。36 台の MD-Board からなる MicroDeep 実行基盤において人物転倒検知実験を実施した結果、太陽光が得られる明るい環境下において環境発電のみで転倒検知判定が実行可能であることを確認した。また、各センサノードの電力消費量を測定した結果、消費電力は最大で 88.4mW であり、明るい屋内における MD-Board の太陽光発電パネルの発電量とほぼ同等であることがわかった。現在、MicroDeep 実行基盤を奈良先端科学技術大学院大学が有するスマートホーム環境に設置し、連続稼働実験を実施している。

2. 関連研究と位置づけ

2.1 エネルギーハーベスティング技術

熱や振動、光を電気エネルギーに変換し、環境に存在するエネルギーを収穫するようなエネルギーハーベスティングは、電源非依存の環境センシングや状況認識を可能にすると期待されている技術である。文献 [5] では周辺の熱や温度変化を利用して発電し、動作する様々なセンサを提案している。文献 [6] は環境発電によって得られた電力量から直接人の行動を推定する技術を提案している。AllSee [7] は環境電波で動作する電波受信機と信号処理ロジックによって周囲の無線の変化を検出し、ユーザのジェスチャを認識するシステムである。これらの環境発電で動作する行動認識は、各センサデバイスでの省電力な行動認識に焦点を当てているため、複数のセンサからのデータを集約して計算処理を行う機械学習には適用できず、我々の目的には適さない。

い。我々は無線センサネットワーク上での環境発電によるニューラルネットワークの実行を目指している。ニューラルネットワークを個々のセンサノードに分散させることで、各ノードが実行する処理は他ノードとの単純なデータ送受信となり、各ノードが環境発電で動作することを可能とする点で異なる。

2.2 モバイル端末上での深層学習

近年では、モバイル端末による深層学習に関する研究もなされている。MoDNN [8] は、事前に学習した CNN モデルをモバイル端末上に分割することによって、CNN の高速化を行う手法を述べている。DeepX [9] や DeepMon [10] では、複数のコプロセッサで計算処理を分割することによって、モバイル端末上での CNN の実行が可能であることを示している。これらのアプローチでは、事前に学習した CNN をモバイル端末上で実行することが可能であるが、環境発電レベルでの電力で動作する端末を考慮したものではない。

2.3 本研究の位置づけ

ニューラルネットワークの分散実行に関する研究はされているが、それらはいずれも膨大なデータ量を迅速に処理するために複数の計算機を用いる並列分散計算である。これに対し、本研究では IoT センサネットワークによる深層学習の新しいフレームワークを提案している。提案手法では、高性能な GPU を備えたサーバーではなく、計算資源の少ない IoT センサノードを協調・連携させることにより負荷の高い学習をセンサネットワークで実現する点でこれまでのアプローチとは全く異なる。また、環境発電のみでニューラルネットワークを実環境で分散実行させた報告は我々の知る限りなく、先駆性を有するアプローチであると考えられる。

3. 分散機械学習プロトコル MicroDeep

本章では、本研究において用いる CNN の分散実行処理のベースプロトコルとなる MicroDeep について述べる。MicroDeep では、各センサノードを CNN のユニットに対応付けている。順伝播処理では、各センサノードは WSN (無線センサネットワーク) を経由して、ユニットの出力データを交換しながら処理を行う。逆伝播処理に関しては本研究における MicroDeep 実行基盤の実装では用いていないため割愛する。詳細は文献 [4] を参照されたい。

3.1 ユニットの割当て

WSN 間で、任意のセンサノード同士の通信が可能であり、通信コストを考慮しないのであれば、任意のユニット割当てで CNN の実行は可能であると考えられる。しかし実際のところ、ユニットの割り当て方法によって、センサノー

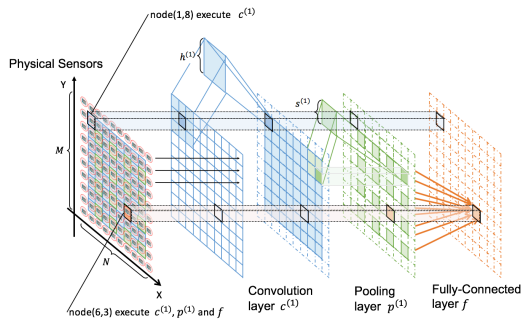


図 1 センサノードのユニット割当て

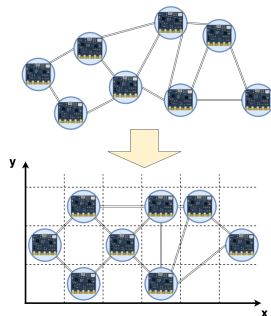


図 2 WSN の座標系への割当て

ド間の通信コストや CNN 実行のオーバーヘッドは変化するため、最適な割当てを行うことが望ましく、処理不可や通信負荷を均等に振り分けるユニットの割当てが重要である。

理想的なセンサノードへのユニット割当ては、WSN のノード間リンクと CNN のユニット間リンクとの対応を最大化し、各センサノードに割り当てられるユニットの数を可能な限り平均化することである。このような割当てでは、特定のノードに過負荷を課すことなく、CNN を実行するためのほとんどの通信が隣接するノード間で行われる。しかし実際のところ、WSN のリンクが CNN のリンクに必ずしも完全対応するわけではないため、そのような割当てが存在するとは限らない。また、CNN と WSN 間の対応リンクを最大化する割当てを探す問題は、部分グラフ同型判定問題に帰着され、最適解を求めるのは NP 困難問題であるため、そのような最適解を求めることは容易ではない。

したがって、MicroDeep では 2 つのセンサノード間の物理的距離を WSN におけるそれらの間の無線リンクに対応させることによって、センサノードの座標を CNN の XY 座標に適用させている。具体的には、各センサノードの位置情報に基づいて、CNN の XY 座標平面に割り当てる簡単な手法を用い、 (x, y) の CNN の各ユニットを (x, y) のセンサノードに割り当てる。理想のシナリオは、 $N \times M$ 個のセンサノードがグリッド状に配置され、センサノードが XY 座標平面に 1 対 1 のマッピングが可能であり、CNN のすべてのユニットをそれらのセンサノードに割り当てることができる状態である。しかし実環境では、配置の制約によりグリッド状にセンサノードを配置できず、図 2 に示す

ようにすべての座標平面にセンサノードを割り当てられない状況が想定される。

このような状況下において MicroDeep では、センサノードが割り当てられない座標を含んだ状態でグリッド状に分割する。センサノードが割り当てられなかった座標はセンサノードの欠損として、一部データが欠けた面のデータとして CNN のユニット処理を行う。センサノードがグリッド状に割り当てられないシナリオとして、図 2 のケース以外には、センサノードの配置に偏りがあり 1 つの座標に 2 つ以上のセンサノードが割り当てられる場合が考えられる。この場合は、グリッド状になるように適切に分割し、1 座標に対し 1 つのセンサノードが割り当てられるようにする。この割当て法により、すべてのセンサノードのセンシングデータを利用できるという利点があるが、配置の偏りの程度によっては、欠損ノードが発生する可能性があるため、先述したような処理を行う。

各センサノードが XY 座標に割り当てられると、図 1 に示すように座標 (x, y) のセンサノードは、CNN においてその座標系に割り当てられた各層のユニットの処理を行う。

3.2 順伝播処理

説明の簡略化のため、WSN 内の $N \times M$ 個のセンサノードが $N \times M$ のアドレス空間に割り当てられているとする。つまり、センサノードとユニットが 1:1 に対応しており、欠損ノードがない状態である。また、各センサノードは他のセンサノードの座標およびセンサノード間のリンクが既知であるとする。これらの情報を基に MicroDeep では、隣接ノードから順番に幅優先探索を実行し、送信先のノードへの最短送信経路を探索する。探索の際にはランダムに隣接ノードを選択することで、データ送信経路の集中による特定ノードの負荷増大を抑制する。

MicroDeep の順伝播処理は、センサノード間の通信によって、通常の CNN と同様に行われる。具体的には、センサノード (x, y) は、割り当てられた入力層のユニット (x, y) 、 k 番目のフィルタの t 番目の畳み込み層のユニット $c_k^{(t)}(x, y)$ 、 t 番目の畳み込み層の後のプーリング層のユニット $p^{(t)}(x, y)$ 、全結合層のユニット $f(x, y)$ 、出力層のユニット $o(x, y)$ を実行する。すなわち CNN において (x, y) に対応するすべてのユニットは、センサノード (x, y) で実行される。

畳み込み層のユニット $c_k^{(t)}(x, y)$ を実行するために、センサノードは前の層 (入力層、畳み込み層、プーリング層等) のユニットから入力データを取得する。そのために MicroDeep では、オフセット O_x, O_y を導入する。これは周囲のノードからデータを取得する際に、そのノードまでの距離を表したものである。オフセット O_x, O_y はプーリング処理を行う度に以下の式によって加算される。 $s^{(t)}$ は $p^{(t)}$ のプーリングサイズを示す。

$$(O_x^{(t)}, O_y^{(t)}) = (O_x^{(t-1)} + s^{(t-1)}, O_y^{(t-1)} + s^{(t-1)}) \quad (1)$$

なお、前の層にプーリング層が存在しなければ、 $s^{(t-1)} = 0$ とする。畳み込み層のユニット $c_k^{(t)}(x, y)$ は、前の層のユニット $p^{(t-1)}(x + i \cdot O_x, y + j \cdot O_y)$ に対応するセンサノードと通信する。ただし、 i, j は $\underline{p}_k^{(t)} \leq i, j \leq \bar{p}_k^{(t)}$ とし、 $\underline{p}_k^{(t)}, \bar{p}_k^{(t)}$ は

$$\underline{p}_k^{(t)} = -\lfloor \frac{h^{(t)}}{2} \rfloor, \quad \bar{p}_k^{(t)} = \lfloor \frac{h^{(t)}}{2} \rfloor - d \quad (2)$$

とする。 $h^{(t)}$ は $c_k^{(t)}$ のフィルタサイズを示し、 $h^{(t)}$ が奇数の場合、 $d = -1$ 、偶数の場合は、 $d = 0$ である。

そして、以下の式によって畳み込み処理が行われる。 f は活性化関数 (ここでは ReLU) であり、センサノード (x, y) での重みおよびバイアスをそれぞれ、 w_{ij} および b とする。

$$c_k^{(t)}(x, y) = f \left(\sum_{\underline{p}_k^{(t)} \leq i, j \leq \bar{p}_k^{(t)}} w_{ij} \cdot p_{ij} + b \right) \quad (3)$$

ただし、 $p_{ij} = p^{(t-1)}(x + i \cdot O_x, y + j \cdot O_y)$ とする。入力データとして、前の層にプーリング層が存在しない場合は、入力層および畳み込み層のデータを用いて計算される。

プーリング層では、以下の式によってプーリング層のユニット $p^{(t)}(x, y)$ の処理が行われる。

$$p^{(t)}(x, y) = \max_{\underline{p}_k^{(t)} \leq i, j \leq \bar{p}_k^{(t)}} c_{ij} \quad (4)$$

ただし、 $c_{ij} = c_k^{(t)}(x + i \cdot O_x, y + j \cdot O_y)$ とし、 $\underline{p}_k^{(t)}, \bar{p}_k^{(t)}$ の導出の際には、フィルタサイズ $h^{(t)}$ の代わりに、プーリングサイズ $s^{(t)}$ を用いる。

3.3 ノード欠損時の処理

本章のこれまでの設計は、 $N \times M$ の座標すべてにセンサノードが割り当てられていることを前提としている。しかし、実環境での実装を考えるのであれば、必ずしもすべての座標にセンサノードが割り当てられているとは限らない。例えば、センサノードが 15 個しか配置されていない場合、 4×4 の座標平面にすべてのノードを割り当てることはできない。また、何らかの要因によりセンサノードが正常動作しない状況も考えられる。このようにセンサノードが欠損している場合においても、順伝播処理を実現するために、MicroDeep では代替処理を行う。

センサノードの欠損は、CNN の入力データの一部が存在しないことを意味する。この問題を解決するために、ユニットは入力データの欠損部分を 0 で補間する。こうして補間した入力データは、畳み込み処理の線形結合性により無視される。加えて、センサノードの欠損は、畳み込み層の出力結果を失うことと同義である。そのため、畳み込み層の出力を必要とする場合においても、ユニットは 0 で補間を行う。この補完したデータも同様に、次の畳み込み層または、

プーリング層の処理によって無視される。畳み込み層では上述した線形結合性、プーリング層では最大プーリング処理によりそれぞれ 0 という値は無視される。プーリング層において、ユニットが欠損した場合も同様である。

全結合層および出力層の処理においては、1 つのノードにデータを集約する必要があるため、集約するノードが欠損した場合は順伝播処理を実行することができない。その場合は、複数ノードでそれらの処理を並列実行する冗長化などの対応が考えられる。

4. MicroDeep 実行基盤の設計開発

3 章で解説を行った分散学習プロトコル MicroDeep をエナジーハーベスティングな WSN へ実装するため、各センサノードとして、センサとマイクロプロセッサおよび太陽光発電機で構成される MD-Board を設計および実装した。以下に MD-Board の設計開発と複数台の MD-Board を連携させた MicroDeep 実行基盤の実装について述べる。

4.1 MD-Board アーキテクチャ

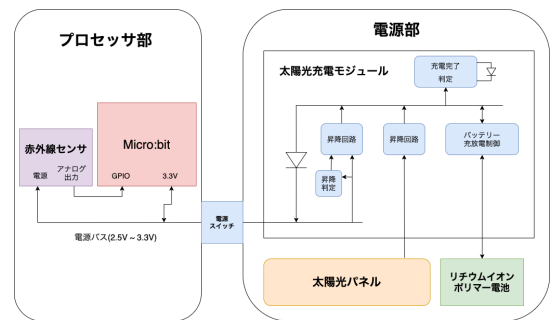


図 3 MD-Board のアーキテクチャ図

本節では、MD-Board のアーキテクチャについて述べる。その構成を図 3 に示す。

MD-Board は、低消費電力なマイクロプロセッサを搭載した micro:bit version 2 [11] と焦電型赤外線センサ (人感センサ) を含むプロセッサ部、および太陽光で環境発電し電源供給を行う電源部から構成される。複数のセンサ値を基に CNN を用いて環境内の人物の転倒を検知するシナリオを通して評価を行うため、焦電型赤外線センサを環境センサとして搭載している。プロセッサ部と電源部は、電源スイッチを介して電源バスと接続しており、電源スイッチが OFF の状態では、プロセッサ部は USB コネクタ等の外部電源からの給電によって動作する。電源部には、3.7V、400mAh のリチウムイオンポリマー電池と太陽光パネルが搭載されている。

以降、MD-Board の各構成機器の詳細について述べる。

4.1.1 センサノード micro:bit

WSN の省電力化を最大限に行うために、可能な限り低消費電力なプロセッサをセンサノードとして用いること

表 1 micro:bit version2 の仕様

| 項目 | 詳細 |
|--------|---|
| プロセッサ | Nordic Semiconductor nRF52833 (ARM Cortex-M4) |
| メモリ | 512kB フラッシュ ROM, 128kB RAM |
| 無線通信規格 | BLE, Bluetooth5.0 / 2.4Ghz micro:bit Radio |
| センシング | 3軸加速度計, 温度センサ, 磁気センサ |
| 出力装置 | グリッド状の5×5赤LED |

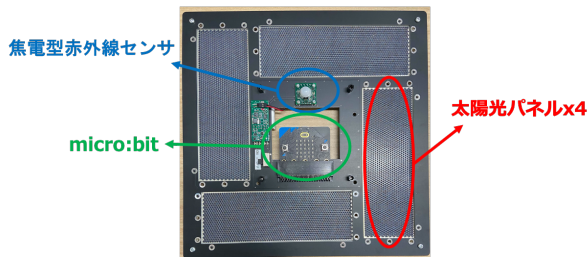


図 4 MD-Board

が重要である。加えて、MicroDeep 実行のための処理能力やメモリ容量、無線通信機能等を備えている必要がある。そこで本研究では、センサノードとして英 BBC 製の micro:bit version2 を用いる。使用する micro:bit の仕様を表 1 に示す。

本研究では、MD-Board36 台を 6×6 のグリッド状に配置し、CNN の分散実行処理を行う。

4.1.2 太陽光パネル

MD-Board は図 4 のように右巴に 4 枚の太陽光パネルが配置されている。これに太陽光を当てることでリチウムイオンポリマ電池が充電される。パネルには、最大 750mW の起動電力を持つ球状単結晶のパネルを用いる。

満充電時には micro:bit と赤外線センサから構成されるプロセッサ部をおよそ 10 時間動作させることが確認できた。

4.1.3 焦電型赤外線センサ

人物の動きの変化を捉え、得たセンサデータを MicroDeep 実行の際の入力データとして扱うため、MD-Board のプロセッサ部に焦電型赤外線センサを搭載している。焦電型赤外線センサは、感知範囲内の熱源の変化を電圧の変動としてアナログ出力する。具体的には、感知範囲内の熱源の配置に変化がない場合に一定の電圧を出力するが、人が通過すると出力電圧が大きく変化する。その後、変化した電圧は時間経過とともに定常時の電圧に収束する。出力のアナログ電圧は GPIO を通して micro:bit 側で取得し活用することが可能である。

4.2 MicroDeep の実装

本節では micro:bit を用いた、MicroDeep の実装方法について述べる。

micro:bit は micro:bit radio という独自の通信プロトコルを用いて通信を行う。micro:bit radio はコネクションレス型のシンプルな通信プロトコルであり、最大 251B プラットフォーム識別用のプリアンブル付きのメッセージをブロードキャスト送信することが可能である。micro:bit

radio では送信側と受信側で 2.4GHz を基準として 1MHz の幅でチャンネルを設定することができ、同一チャンネル経由で受信したメッセージのみが受信メッセージキューに入る仕組みである。しかし、micro:bit radio はコネクションレス型であるため、メッセージが受信される保証がなく、受信できた時に応答を返すことができない。

実装した MicroDeep では、micro:bit radio の性質を考慮し、CNN の各層の処理をフィルタ適応やプーリング処理の単位にグループ化させ、グループ毎の処理を逐次実行していくことで順伝播処理を実現する。以降でグループ毎の処理方法とグループ間での同期方法について述べる。

4.2.1 ノードグループ内の通信手法

先述したように micro:bit radio の通信はコネクションレス型であるため、メッセージが受信される保証がなく、受信できた時に応答を返すことができない。この性質を考慮すると、畳み込みやプーリング処理において送受信を行いつつのノードにデータを集約する方法としては、まずノード全体の時刻を定期的に同期させ、送受信のタイミングが一致しないようにタイミングを制御することが考えられる。しかし同期を行うための通信が必要となり、限られたメモリ容量や電力制約のもとでは望ましくない。したがって、提案手法では、各層の処理単位のユニットに対応するノードグループを形成する。そして、グループ単位で個別のチャンネルに移動させ、データの送受信を行う。このとき、一定時間内で送受信処理を繰り返し実施させることでデータ欠損の影響を軽減する。データが受信できなかった場合は前述のように欠損したノードのデータを 0 で補完する。

4.3 ノードグループ間の制御方法

グループ分けを行なったグループ単位の実行の方式として、並列的な実行と逐次的な実行が考えられる。

並列的な実行を行う場合、CNN の畳み込みとプーリングの処理において、フィルタサイズ、プーリングサイズ、ストライド数等の要因により、図 5 に示すように複数のノードグループに属すノードが生じてしまう問題がある。この際、複数のグループを同一のチャンネルに設定し並列的に実行することは可能ではあるが、micro:bit radio の送信処理は設定チャンネルに対しブロードキャストで行われるため、隣接しないノードからのメッセージが受信ノードの受信メッセージキュー領域を占めてしまうという問題がある。また、同一チャンネル内で送受信処理を同時に行うノード数の増加に伴い、データの衝突が起きやすくなる恐れがある。

一方で、図 5 のような場合においては、一定周期でチャンネルを切り替えるような割り込み処理を加え、複数のノードグループに対して送信するという方法も有用である。しかし、図 6 のような送信処理と受信処理を同時に実行するノードが発生する場合を想定したとき、チャンネルの切り替えに加えデータを相互に交換できるような同期処理が必要

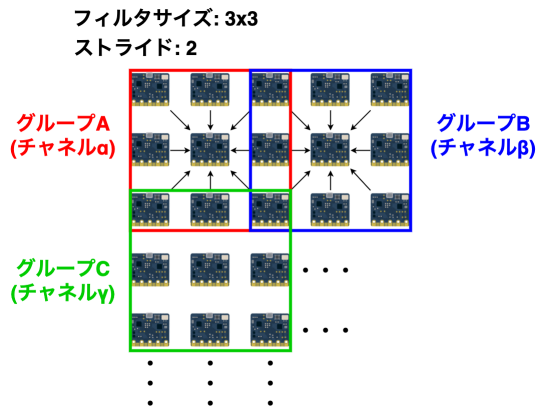


図 5 ノードが複数のグループに属す例 (ストライド 2)

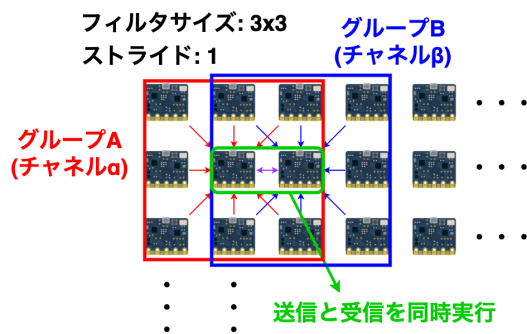


図 6 ノードが複数のグループに属す例 (ストライド 1)

となるが, micro:bit のような計算資源やメモリ容量が限られたマイコンボードで上記のような様々な局面に対して最適な処理を行う実装は容易ではない. 本研究では様々な CNN に対して汎用的に処理を行うことが可能な逐次的な処理方式を用いる.

複数のノードグループを逐次的に処理するためには, 各センサノードの動作をメッセージで自由に制御するような仕組みが必要である. しかし, micro:bit radio では複数のチャンネルを同時に所持することはできないため, メッセージを送信する場合は常に受信側の受信チャンネルを把握しておかなければならないが, 自身以外のセンサノードのチャンネル移動を常に管理し続けるのは容易ではない. これに対し, グループ処理を終えたノードが次に処理を開始するグループに制御メッセージを送信する動作を繰り返し行うことで, チャンネル管理を用いずに先述した逐次的な処理を実現する. 具体的には, 処理を行っていないノードが留まる全体用チャンネルを用意し, グループ単位の処理を終えたうちのあるノードが全体用チャンネルに復帰した際に次グループの開始を示す制御メッセージを送信することでメッセージ衝突を回避する.

5. 動作検証

5.1 検証内容

本章では, 実装したセンサノード MD-Board を 36 台連

携させた MicroDeep 実行基盤上で, 全 MD-Board の人感センサからのデータを用いた転倒検知用の軽量 CNN を分散実行できることを確認する. 以下に述べる想定シナリオにおいて取得した実データを入力値として学習させた CNN の重みおよびバイアスを実装した MicroDeep 実行基盤に与えることで, 同実行基盤上での判定処理を実施する. この処理中に消費される各センサノードの電力を計測し, MicroDeep を環境発電で実行するために必要な環境条件についての考察を行う.

5.2 想定シナリオ

本稿では, WSN 活用シナリオとして異常行動の検知を考える. 圧力検知が可能なセンサアレイを内蔵したフロアマットや, アレイ状に構成された焦電型赤外線センサを用いた広範囲モーションセンサ等に, 家庭内の高齢者の行動パターンを学習させることで, 高齢者の転倒検知や侵入者の検知などが可能となる. これらはカメラやマイクなど侵襲性の高いデバイスを用いる必要がなく, プライバシー保護の観点からも有用である.

本研究では焦電型赤外線センサを用いた転倒検知を行う CNN を構成した. 具体的には, 4 章で述べた焦電型赤外線センサ付きの MD-Board を 36 台実装し, 壁面に 6×6 のアレイ状に配置し, MicroDeep による CNN 処理を用いて転倒とそれ以外の二値分類判定を行う.

5.3 学習データとネットワーク構成

MD-Board 上で転倒検知を行うための CNN は事前学習を行なったモデルパラメータを用いる. 学習に用いるデータセットは, あらかじめ 6×6 のアレイ状に配置した焦電型赤外線センサで毎秒 5 サンプルのサンプリング周期で収集された, 5 人の被験者による計 55 回の歩行データである. その内, 23 回が異常のラベル付けがされた転倒のデータ, 残りの 32 回が正常のラベル付けがされた正常歩行のデータである. 本稿においては, 転倒を検知するのに必要な計測時間の長さは 2 秒と仮定し, 10 サンプル分を 1 つのデータとして扱う. すなわちネットワークの入力は $6 \times 6 \times 10$ の 2 次元配列データとなる. なお, 計 55 回の歩行データは学習には不十分であるため, 1 サンプルごとにスライディングさせるデータ増強手法を適用することで, 1600 回分のデータを生成するデータ拡張を適用した. これらのうち, 530 個を異常データ, 残りの 1070 個を正常とラベリングした.

転倒検知のために, 畳み込み層 1 層, プーリング層 1 層, 全結合層 1 層から構成されるネットワークを作成し, 学習を行なったところ, 84% の精度となった. MD-Board の micro:bit 上で再現した構成したネットワークに, この学習させたモデルから抽出した重みやバイアスといったパラメータを組み込み, 転倒検知を行う.



図 7 実装した MicroDeep 実行基盤 (36 台の MD-Board および判定結果表示用の MD-Board)

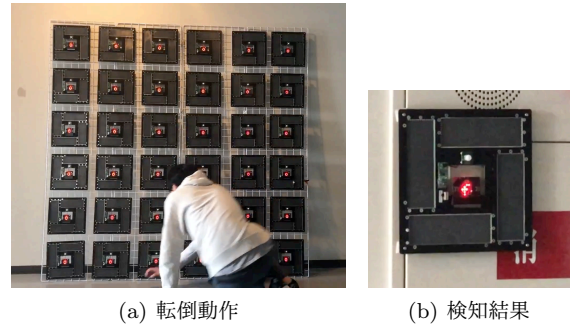


図 9 屋内での転倒検知動作検証の様子 : (a) 転倒動作 (b) 検知結果

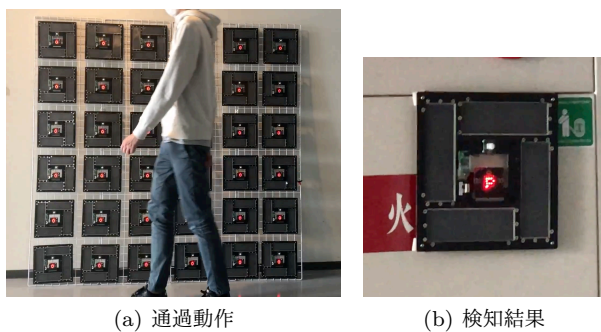


図 8 屋内での転倒検知動作検証の様子 : (a) 通過動作 (b) 検知結果



図 10 冬季屋外での動作検証の様子

5.4 MicroDeep 実行基盤の動作

センサノードである MD-Board36 台を図 7 のようにグリッド状に配置した。ここで、各 micro:bit に焦電型赤外線センサの出力電圧の変動を常時モニタリングさせることは電力消費の観点から望ましくないため、本研究では、CNN を実行する 36 台の MD-Board に加え、7 のようにそれらの横に人の通過を常時モニタリングするトリガ用 MD-Board を設置した。トリガ用 MD-Board が人の通過を検知すると、監視用チャンネルにメッセージを送信する。他の各 MD-Board はセンシング処理と転倒の判定処理を行う。このトリガ用 MD-Board は転倒検知の判定結果の表示にも使用する。MD-Board の CNN が正常行動を検知した場合には“pass”を、転倒を検知した場合には“fall”のメッセージをトリガ用 MD-Board に送信し、トリガ用 MD-Board は受信メッセージを自身の LED 上でスクロール表示する。

5.5 検証結果

まず実装した WSN で実際に転倒検知が来ているか確認するために、検知範囲内で転倒と通常歩行を複数回行った。その様子を図 8(a) - 9(b) に示す。図 8(a) と図 8(b) では、通過時にトリガ用 MD-Board の LED に“pass”と表示されており、MicroDeep 実行基盤が正常行動を検知できていることが確認できる。また、図 9(a) と図 9(b) では、通過

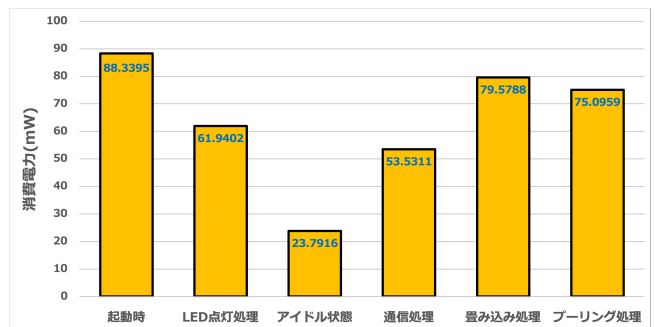


図 11 様々な実行状況下での MD-Board(プロセッサ部)の最大消費電力

時にトリガ用 MD-Board の LED に“fall”と表示されていることから MicroDeep 実行基盤が異常行動を検知できていることも確認できる。

次に、様々な実行状況における MD-Board あたりの消費電流ならびに電圧を測定した。マイクロアンペアオーダーでの高精度測定が可能な HIOKI デジタルマルチメータ DT4282 を用いた計測の結果、電流は micro:bit の処理に応じて大きく変動した。一方電圧の変動は微小であり、転倒検知処理全体を通しての平均値は 2.93 V であった。電流と平均電圧を基に算出した消費電力を図 11 に示す。本稿における実験においては、各 MD-Board の micro:bit の動作に応じて micro:bit が有する 5 × 5 の LED 点灯処理を実行させている (デモ用途)。この消費電力は通信処理と比べて大きく、実環境における運用時には使用すべきではないことがわかる。次に畳み込み層の消費電力がプーリン

グ層の消費電力より大きいことがわかる。これは、畳み込み処理ではフィルタサイズを3としており、9つのノードのデータを集約する必要があるのに対し、プーリングサイズは2であるため、プーリング処理では4つのノードのデータを集約するだけでよいことが理由として考えられる。しかし、いずれの実行処理時の消費電力は起動時の消費電力を下回っていることがわかる。本検証で MicroDeep 実行基盤を設置した屋内は設置可能場所の制約上、太陽光が届かない場所であったためにかかなり暗く、太陽光パネルによる発電容量は MD-Board あたり数 mA 程度であったことから、蓄電池なしでの動作確認は屋内においてはできなかった。しかし、上記の計測結果からは、起動時の電力を賄えるだけの光量が得られる環境下では永続的な動作が可能であることがわかる。また、未動作状態時の環境発電による余剰電力を蓄積することも可能であり、長時間の安定動作が可能であると考えられる。現在奈良先端科学技術大学院大学が有するスマートホームにこの実行基盤を設置し、やや明るい室内における連続稼働性を検証中である。

最後に、環境発電のみによる一連の動作が実行可能かを確認するため、各 MD-Board に内蔵されているリチウムポリマ充電池を物理的に取り外したうえで、2021年2月11日(冬季)午後4時頃の屋外で動作検証実験を行なったところ、センシングから結果表示までの一連の処理を問題なく実行できた。そのときの様子を図10に示す。このことから、冬季夕方方の弱い太陽光であっても、環境発電による動作が可能であることがわかった。

6. おわりに

本研究では、分散型機械学習プロトコル MicroDeep を、環境発電で動作する無線センサネットワーク上に実装した。低消費電力なマイコンである micro:bit、省電力の焦電型赤外線センサ、太陽光電力パネル、リチウムポリマ充電池、ならびに太陽光電力パネルからの供給による蓄電を可能とする電源制御回路を搭載した MD-Board を設計開発および実装し、その MD-Board を36台用いた MicroDeep 実行基盤の動作検証を行った。また、MicroDeep 実行基盤が環境発電による永続的な実行を実現するために必要な条件を確認するための検証実験を実施した。

検証の結果、本研究で実装した個々の MD-Board は起動時に 88.4 mW と最も電力を消費することがわかり、その後はそれ以下の電力で動作することがわかった。したがって、起動時に充電池の電力を利用し、その後は明るい室内で数十 mA の電流供給を受け続けることができれば、永続的に動作すると考えられる。また冬季夕方方の屋外においては、太陽光発電のみで CNN 推論処理を実装した MicroDeep 実行基盤上で実行できることを確認した。

今回は環境発電で実行可能な MicroDeep の順伝播処理のみを構築したが、メモリ運用の最適化や処理の軽量化を

行い、逆伝播処理も同様に実装することが今後の課題である。

謝辞

本研究成果は、国立研究開発法人情報通信研究機構(NICT)の委託研究「未来を創る新たなネットワーク基盤技術に関する研究開発」により得られたものです。

参考文献

- [1] Schwartz, R., Dodge, J., Smith, N. A. and Etzioni, O.: Green AI, *Commun. ACM*, Vol. 63, No. 12, p. 54–63 (online), DOI: 10.1145/3381831 (2020).
- [2] Center for Low Carbon Society Strategy (LCS), J.: Impact of Progress of Information Society on Energy Consumption (Vol.1): Current Status and Future Prospects for Power Consumption of IT Equipment Strategy for Technology Development, *Proposal Paper*, (online), available from (<https://www.jst.go.jp/lcs/pdf/fy2018-pp-15.pdf>) (2019).
- [3] Fukushima, Y., Miura, D., Hamatani, T., Yamaguchi, H. and Higashino, T.: MicroDeep: In-network Deep Learning by Micro-Sensor Coordination for Pervasive Computing, *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 163–170 (online), DOI: 10.1109/SMARTCOMP.2018.00087 (2018).
- [4] 福島悠太, 山口弘純, 東野輝夫: センサネットワークにおける分散型深層学習の設計と評価, *情報処理学会論文誌*, Vol. 60, No. 1, pp. 2–15 (2019).
- [5] Zhao, C., Yisrael, S., Smith, J. R. and Patel, S. N.: Powering wireless sensor nodes with ambient temperature changes, *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2014)*, pp. 383–387 (2014).
- [6] Khalifa, S., Hassan, M. and Seneviratne, A.: Pervasive Self-Powered Human Activity Recognition without the Accelerometer, *Proceedings of the 13th IEEE International Conference on Pervasive Computing and Communications (PerCom 2015)*, pp. 79–86 (2015).
- [7] Kellogg, B., Talla, V. and Gollakota, S.: Bringing Gesture Recognition to All Devices, *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation (NSDI 2014)*, p. 303–316 (2014).
- [8] Mao, J., Chen, X., Nixon, K. W., Krieger, C. and Chen, Y.: MoDNN: Local distributed mobile computing system for Deep Neural Network, *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, pp. 1396–1401 (2017).
- [9] Lane, N. D., Bhattacharya, S., Georgiev, P., Forlivesi, C., Jiao, L., Qendro, L. and Kawsar, F.: DeepX: A software accelerator for low-power deep learning inference on mobile devices, *Proceedings of the 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, IEEE, pp. 1–12 (2016).
- [10] Huynh, L. N., Lee, Y. and Balan, R. K.: DeepMon: Mobile GPU-based Deep Learning Framework for Continuous Vision Applications, *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, ACM, pp. 82–95 (2017).
- [11] Micro:bit Educational Foundation: micro:bit, <https://microbit.org/>. [Accessed 15-February-2021].