

# センサーネットワークにおける分散型深層学習の設計と実装

福島 悠太<sup>1,a)</sup> 三浦 太樹<sup>1,b)</sup> 濱谷 尚志<sup>1,c)</sup> 山口 弘純<sup>1,d)</sup> 東野 輝夫<sup>1,e)</sup>

**概要:** 無線センサーネットワークにおいて、センサーに付随するマイコンの高機能化・省電力化が進めば、従来クラウドで行っていた学習や異常検出、判定などのタスク処理をセンサーネットワークにオフローディングし、データ発生場所に近い場所でそれらを効率よく行える自律的な知能センサーネットワークが実現できる。本研究では CNN を対象に、ローカルな無線センサーネットワーク内で分散実行する新しいアーキテクチャを提案し、そのための分散実行プロトコルならびにアルゴリズムを提案する。提案手法はメッシュ型の無線センサーネットワークが面的かつ定期的に取得するデータを対象とし、センサーノードに深層学習におけるユニットの役割を割り当てる。提案手法の有効性を評価するため、1,400m<sup>2</sup> 超の実ラウンジスペースの 50 地点の温度データを用いて、通常の CNN による学習と提案手法による分散学習におけるデータ通信量と学習精度の比較を行った。その結果、十分妥当な通信量のもとで、通常の CNN と遜色ない学習精度を達成できることがわかった。

## 1. はじめに

分散コンピューティングのフレームワークは、サーバーやクライアントの通信性能や可用性、性能差などに応じ、クライアント指向やサーバー指向へと常に変遷しているが、近年では、クラウドサービスの信頼性と可用性向上により、データ処理をクラウドで集約して行う傾向がある。一方、マイクロソフトや Google TensorFlow などによるいくつかの IoT ツールは、機械学習等により訓練された判定関数などを IoT デバイスに導入できるエッジコンピューティング機能をサポートしつつある。これにより、全センシングデータをクラウドサーバーに送信するための通信リソースを確保する必要がなくなり、プライバシーデータをローカルで処理することも可能となる。しかし、そういった既存のツールおよび既存アプローチのほとんどは、学習済みの判定機能の一部または全部をクラウドからエッジデバイスに移行し、以降の検知や判定をエッジサイドで行うことで、以降のクラウドへのデータ量を削減することが目的である。したがって、学習段階では依然として、学習機能を有するクラウドサーバーあるいはホームゲートウェイなどにすべてのデータを集約する必要がある。

一方、現在 IoT デバイスと呼ばれるセンサーノードは、メモリ量や処理能力という点での性能向上が著しく、センシングやそのデータ送信のみならず、機械学習においても一部のタスクを実行することが可能であると考えられる。現状の無線センサーネットワーク (WSN) は、経路制御を含むデータ集約プラットフォームとみなされることが多いが、これらのノードをシームレスに連携させることで、より多くのメモリと処理能力が利用可能となり、センシング・学習・フィードバックのプロセスのほとんどがセンサーネットワーク内で完結するような、知的な局所型データ処理プラットフォームを形成できる。

そこで本研究では、WSN において深層学習およびそれを用いたデータ処理を行うための新しい手法を提案する。提案手法では、WSN の各センサーノードから連続的に生成されるデータを 2 次元や 3 次元の地理的データ (例えば、温度データ等) として扱う。これに対し、多層ニューラルネットワークの中でも画像認識をはじめとする多くの分野において高い精度を達成できる畳み込みニューラルネットワーク (CNN) を対象とし、CNN の各順伝播ユニット (畳み込み処理 (フィルタ) やプーリング、全結合など) を、WSN 内のノード (センサーノード) のいずれかに割り当てる。また、逆伝播時のパラメータ更新処理を分散型で行う仕組みを開発することで、ユニットのパラメータ更新を分散環境で実現する。通信量やノード処理負荷の偏りをなるべく少なくするユニット割り当てならびにプロトコル設計を行うことで、各ノードの最大処理負荷や最大通信量を抑制しながら分散環境で CNN を実現している。

<sup>1</sup> 大阪大学 大学院情報科学研究科  
Graduate School of Information Science and Technology, Osaka University, Suita, Osaka, Japan

a) y-fukushima@ist.osaka-u.ac.jp

b) d-miura@ist.osaka-u.ac.jp

c) h-takasi@ist.osaka-u.ac.jp

d) h-yamagu@ist.osaka-u.ac.jp

e) higashino@ist.osaka-u.ac.jp

提案手法の学習精度を検証するために、実環境で取得したセンシングデータを用いた実験を行った。具体的には、 $1,400m^2$  超の実ラウンジスペースにおいて 50 個の温度センサーを設置し、スポット温度の異常検知を行うシナリオを対象とした。その結果、提案手法（分散型 CNN）の学習精度は 95.570%，データを WSN で集約して学習を行う方式（通常型 CNN）では 97.095% となった。また、分散型 CNN では一度の学習にかかるセンシングデータ送受信量が最大のノードにおいても 404 データであったのに対し、通常型 CNN では 1,988 データであった。これらから、ノードにかかる最大通信負荷を抑制しながら通常の CNN と遜色ない学習精度を達成できることが確認された。

## 2. 関連研究と位置づけ

近年、深層学習は多層ニューラルネットワークを学習する方法として幅広く研究されており、物体認識 [1]、画像検索 [2] など様々なデータ解析において大きな成果が得られている。しかし、膨大なデータ量により訓練する場合にはデータ量に応じた処理コストが要求される。そこで、訓練にかかる処理負荷の軽減を目指した分散実行手法が提案され、また、それらの手法を容易に扱うことを可能にする Theano [3]、Torch [4]、Decaf [5]、Overfeat [6] and Caffe [7] などの多くのフレームワークが開発されている。

文献 [8] では、大規模な深層学習のモデル学習のために、数千のコアおよび数千の計算スレッドを有する GPU と分散システムを用いる手法を提案している。他に深層学習の並列化として、文献 [9] では 16,000 個の CPU を用いることで数十億のパラメータを持つ大規模な深層学習モデルでの学習精度を向上させる手法が提案されている。また文献 [10] では、6,000 万のパラメータと 65 万のニューロンを持つ大規模な畳み込みニューラルネットワークにおいて、GPU を用いることによる学習高速化が報告されている。

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 [11] 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 [12] や DeepMon [13] 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、モバイル端末上で学習を実行する手法については触れられていない。

前述のように、深層学習の分散実行は多くの研究がなされているが、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 センサー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 センサーノードを協調・連携させることにより負荷の高い学習をセンサー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。

## 3. 提案手法

### 3.1 想定環境とシナリオ

本研究では、ある程度のプロセッシング能力と通信機能を搭載したセンサーノードが無線通信によって WSN を構成している状況を想定している。WSN を  $G$  とすると、センサーノードの集合  $S$  と、センサーノード間の双方向リンク  $E_G \subseteq S \times S$  を用いて、 $G = (S, E_G)$  と表現できる。 $G$  は OLSR などの適切な経路制御プロトコルにより、任意のセンサーノード間には必ず 1 つ以上の通信経路が存在し、互いに通信可能な状態であるとする。本研究では簡単のため、センサーノードは移動しないと仮定するが、ノード間の適切なコネクティビティが経路制御機構により提供されればこれは本質的な制約ではない。また、同一 WSN に所属するセンサーノードがセンサー値の取得タイミングを同期する必要はないが、各ノードは共通の時間間隔でセンサー値を取得し続けるものとする。

WSN 内で深層学習を行う典型的な例として、オフィスなどの屋内空間において温湿度や輝度などをセンシングし、省エネルギーや快適性向上のために適切な空調・照明を提供する BEMS システムがあげられる。特に個人の温熱快適性は、室温や入射日光、エアコンの位置など様々な要因に影響を受ける主観的な値であるため、温熱快適性を実現するように空調の調整を行うことは容易でない [14, 15]。実際に、6 章で用いた実験データからは、対象ラウンジスペース内の最大温度差が 3 度から 4 度程度であったことがわかっており、温熱的な不快感を訴える滞在者も存在する。しかし、前述のように、どのような温度環境が人の快・不快を決定するかがわかりにくいいため、例えば温熱的不快感のクレームの有無や顧客サーベイを真値とし、WSN が快・不快を判断できるようになれば、不快状況をいち早く検知して管理者に通知する用途にも適用可能である。また、センサーアレイにおける異常行動検知なども考えられる。例えば、圧力センサーアレイを内蔵したフロアマットや、安価な焦電型赤外線センサーをアレイ状に構成した広範囲モーションセンサーで、家庭内の高齢者の行動パターンや店舗の通行パターンなどを学習しておけば、カメラなどのプライバシーに関わるデバイスを利用することなく、転倒検知や侵入検知などの異常検知に適用できる。センサーノード間でこういった学習処理を分担することにより、各センサーノードの処理量は十分に小さくなり、圧力発電のようなエネルギーハーベスティングでの処理が可能となる。すなわち、学習処理全体がエネルギーハーベスティングで実行可能となり、省エネルギーで知的な WSN の実現につながることになる。

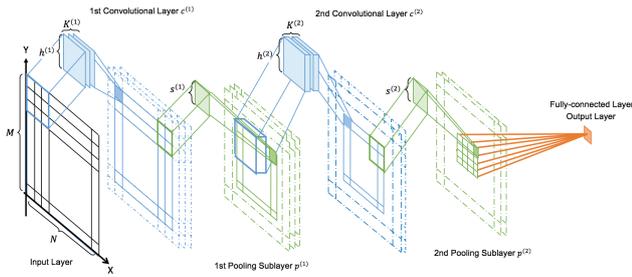


図 1 CNN の例

### 3.2 分散型 CNN

提案手法では、入力層、 $T$  層の隠れ層 ( $1 \leq T$ )、全結合層  $f$ 、出力層  $o$  からなる一般的な CNN を対象とする。  $t$  層目の隠れ層 ( $1 \leq t \leq T$ ) は畳み込み層  $c^{(t)}$  およびプーリング層  $p^{(t)}$  からなり、プーリング層は任意であるものとする。各畳み込み層  $c^{(t)}$  のフィルタ数およびフィルタサイズを、それぞれ  $K^{(t)}$  および  $h^{(t)}$  で表し、プーリング層がある場合のプーリングサイズは、 $s^{(t)}$  で表す。各層のユニットは  $XY$  座標を用いて、ユニットの座標を非負の整数値で  $(x, y)$  のように表し、左下のユニットを原点  $(0, 0)$  とする。  $k$  番目のフィルタの  $t$  層目の畳み込み層  $c^{(t)}$ 、 $t$  層目のプーリング層  $p^{(t)}$ 、全結合層、出力層のユニットをそれぞれ、 $c_k^{(t)}(x, y)$ 、 $p^{(t)}(x, y)$ 、 $f(x, y)$ 、 $o(x, y)$  と表す。畳み込み層、プーリング層は前の層のフィルタサイズ数  $(h^{(t)})^2$  および  $(s^{(t)})^2$  分の出力から、全結合層および出力層は前の層の全ての出力から計算される。

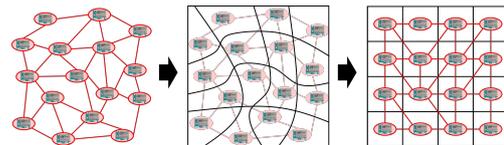
以降、入力層、畳み込み層、プーリング層、全結合層、出力層のサイズをそれぞれ  $N \times M$ 、 $X_c^{(t)} \times Y_c^{(t)}$ 、 $X_p^{(t)} \times Y_p^{(t)}$ 、 $X_f \times Y_f$ 、 $1 \times 1$  で表す。  $X_c^{(t)}$  は  $X_c^{(t)} = X - (h^{(t)} - 1)$  で、 $X_p^{(t)}$  は  $X_p^{(t)} = \lceil X/s^{(t)} \rceil$  で計算される ( $Y$  も同様)。図 1 に  $T = 2$  の場合の CNN の例を示す。

詳細は後述するが、提案手法においては多地点センシングデータを画像のような 2次元データとみなし、各データを有するセンサーノードが周辺のセンサーノードから畳み込みなどに必要なデータを受け取り CNN の順伝播を実現する。したがって、フィルタサイズ  $h^{(t)}$  およびフィルタ数  $K^{(t)}$  が通信量に影響を与える。例えば、フィルタサイズ  $h^{(t)}$  が大きいと、畳み込み処理を行うためには、各ノードがより離れたセンサーノードからデータを集める必要があり、通信量が増大する。したがって、提案手法ではフィルタサイズ  $h^{(t)}$  は比較的小さい値 (3 あるいは 5 程度) が望ましく、本稿で行った実験においては、そのような比較的小さいフィルタが適切であった。なお、一般には最適なフィルタサイズはデータセットに依存する。

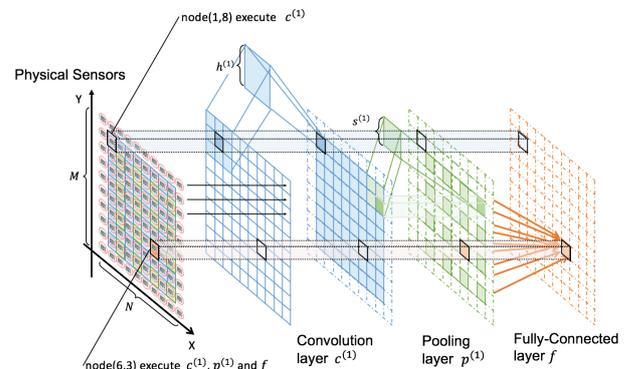
本稿で用いる CNN のパラメータ等の表記を表 1 に示す。

表 1 CNN のパラメータ表記

パラメータ	詳細
$(N, M)$	入力データサイズ
$T$	隠れ層の数
$c^{(t)}$	$t$ 層目の畳み込み層
$p^{(t)}$	$c^{(t)}$ の後のプーリング層
$c_k^{(t)}(x, y)$	$(x, y)$ 座標の $k$ 番目のフィルタの $c^{(t)}$ のユニット
$p^{(t)}(x, y)$	$(x, y)$ 座標の $p^{(t)}$ のユニット
$K^{(t)}$	$c^{(t)}$ のフィルタ数
$h^{(t)}$	$c^{(t)}$ のフィルタサイズ
$s^{(t)}$	$p^{(t)}$ のプーリングサイズ

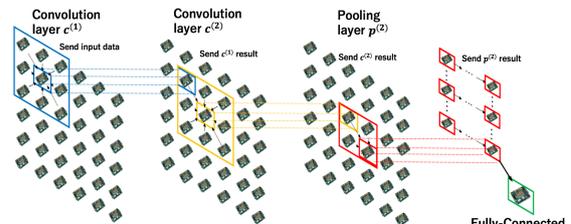


(a) WSN の座標系への割当

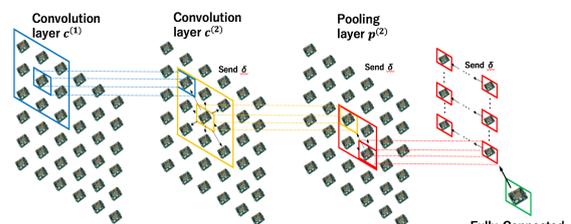


(b) CNN の座標系への割当

図 2 センサーノードのユニット割当



(a) 順伝播



(b) 逆伝播

図 3 分散型 CNN の処理例

## 4. 分散実行プロトコル

### 4.1 動作概要

提案手法では、CNN の各ユニットがセンサーノードに

対応づけられる。順伝播では、各センサーノードは WSN を介して、(入力層を含む) ユニットの出力データを交換し、ユニットの処理を行う。出力層のユニットを割り当てられているセンサーノードは、出力データと正解データから逆伝播処理を行う。逆伝播では、各ユニットが前のユニットの伝播のみに基づいて分散実行し、重みを更新する。したがってユニット間での重み共有は行わず、その更新はユニット毎に独立して行われる。

図 3 に、畳み込み 2 層、プーリング 1 層の場合における CNN の順伝播と逆伝播の処理を示す、WSN 上でこの処理を行うために、(1) センサーノードへのユニットの割り当て、(2) 順伝播におけるユニットの実行、(3) 逆伝播におけるユニット毎の重み更新、の処理が行われる。これらの各処理について次節以降で述べる。

#### 4.2 ユニットの割り当て

WSN 間での任意のセンサーノード間の通信が保証されており、通信量に制限がないのであれば、どのようなユニット割り当てでも CNN の学習は可能である。しかし、ユニット割り当てによって、各センサーノードの通信および処理オーバーヘッドは大きく変化する。例えば、入力層を除いた全てのユニットを 1 つのセンサーノードに割り当てたとすると、このノードに送信されるセンシングデータは、センサー数  $(N \cdot M - 1)$  に比例することに加え、1 つのノードで順伝播および逆伝播のすべての処理を行う必要がある。我々のアプローチはこの処理負荷や通信負荷をなるべくノード均等に振分けるように、割り当ておよび分散型 CNN を実現することである。ここで、理想的なセンサーノードへのユニット割り当ては、CNN のユニット間リンクができるかぎり WSN のノード間リンクに対応し、かつ各センサーノードのユニット割り当て数ができるかぎり平均化される状態である。ユニット間リンクはセンサーノード間のデータ伝送を意味し、その 2 ノードのホップ数は少ないほうが望ましい。特に、直接のリンクがなければ、マルチホップ通信で実現する必要があるため、ネットワークを流れるデータ量が増加する。しかし、WSN のリンクが CNN のリンクに完全に対応するわけではないことから、そのような割り当てが存在するとは限らず、また、CNN と WSN でできるだけ対応リンクが多い割り当てを発見する問題は、部分グラフ同型判定問題 [16] に帰着され、最適解を求めるのは NP 困難問題とされている。

したがって、本研究では、CNN の局所性を利用したヒューリスティックな割り当て手法を採用する。具体的には、CNN では近隣のデータを畳み込み層およびプーリング層の入力に用いることが多いため、互いに隣接する 2 つのセンサーノードに CNN の隣接する座標が割り当てられるように、センサーノードを CNN の正方座標に割り当てる。これにより、WSN の各センサーノードに座標が与え

られ、CNN のリンクを WSN のリンクでカバーできる割合をできるだけ高くするような CNN の座標系に写像できる。

具体的には、アドレス変換関数として  $\mathcal{T}: S \rightarrow N \times M$  という全単射関数を用いる。 $(x_s, y_s)$  は座標  $\mathcal{T}(s)$  を意味し、 $\mathcal{X}_c^{(t)}$  と  $\mathcal{Y}_c^{(t)}$  はそれぞれ畳み込み層  $c^{(t)}$  のオフセットを表す。オフセットは、各層のアドレス空間を調節するのに用いるもので、通常、入力層の座標  $N \times M$  より小さい。したがって、 $c_k^{(t)}(x, y)$  のセンサーノードは、アドレス変換関数  $\mathcal{T}$ 、オフセット  $\mathcal{X}_c^{(t)}$  と  $\mathcal{Y}_c^{(t)}$ 、畳み込み層  $c^{(t)}$  を用いて、 $\mathcal{T}^{-1}(x + \mathcal{X}_c^{(t)}, y + \mathcal{Y}_c^{(t)})$  と表される。プーリング層  $p^{(t)}$ 、全結合層  $f$  および出力層  $o$  についても上記の議論を適用することができる。以下では表記の簡単のため、上記の割り当てを  $\mathcal{T}$  とオフセットからなる  $\mathcal{A}$  を用いて、 $\mathcal{A}: U \rightarrow S$  とする、実際には、 $\mathcal{A}$  つまり  $\mathcal{T}$  およびオフセットには多くの選択肢が存在する。そこでグリーディ法を用いて通信および計算オーバーヘッドを最小化するようなセンサーノードの座標を決定する。そのために以下のメトリクスを導入する。

$$HS = \frac{\sum_{(u,u') \in E_N} \text{hop}(\mathcal{T}^{-1}(u), \mathcal{T}^{-1}(u'))}{|E_N| - \sum_{(u,u') \in E_N} \text{isSame}(\mathcal{T}^{-1}(u), \mathcal{T}^{-1}(u'))} \quad (1)$$

$\text{hop}(s, s')$  は  $s$  と  $s'$  間のルートにかかるホップ数 ( $s=s'$  ならば 0) で、 $\text{isSame}$  は  $s=s'$  ならば 1、それ以外は 0 を返す関数である。

$$UD = \frac{\max_{s \in S} \sum_{u \in U} \text{isAssigned}(u, s)}{\text{ave\_unit\_num}} \quad (2)$$

ここで、

$$\text{ave\_unit\_num} = \frac{N \cdot M + \sum_{1 \leq t \leq T} \{K^{(t)} \cdot \mathcal{X}_c^{(t)} \cdot \mathcal{Y}_c^{(t)} + X_p^{(t)} \cdot Y_p^{(t)}\} + 1}{|S|}$$

$\text{isAssigned}$  は  $\mathcal{A}(u) = s$  ならば  $\text{isAssigned}(u, s) = 1$ 、それ以外なら 0 となる関数で、 $\text{ave\_unit\_num}$  は各センサーノードに割り当てられているユニットの数の平均値である。

$HS$  は、オーバーレイネットワークと物理ネットワークのリンク一致性を表し、 $\text{Hopstretch}$  [17] と呼ばれるメトリクスである。理想的には、上記の値が 1 となることであり、それはオーバーレイネットワークのリンクがすべて物理リンクで構成されることを意味する。極端な例では、WSN がフルメッシュであれば  $HS$  は 1 となる。

$UD$  はユニット割り当て偏差であり、センサーノードに割り当てられたユニットの最大数と平均の比率を表す。 $UD$  の値が小さいほどセンサーノードへのユニットの割り当てが均一的であるということの意味する。

そのような理想に近い  $HS$  および  $UD$  を有する割り当てのため、我々は、センサーノード間の通信関係をできるだけ保つようにセンサーノードをグリッド状に割り当てる方法を採用する。WSN では通常シンクノードと呼ばれるノードが、各ノードのホップ数を得ることで完全なトポロ

ジ情報を知ることができる。これを用いて、以下の手法を用いる。

- (1)  $\mathcal{X}_c^{(t)} = \frac{N-X_c^{(t)}}{2}$ ,  $\mathcal{Y}_c^{(t)} = \frac{M-Y_c^{(t)}}{2}$ ,  $\mathcal{X}_p^{(t)} = \frac{N-X_p^{(t)}}{2}$ ,  
 $\mathcal{Y}_p^{(t)} = \frac{M-Y_p^{(t)}}{2}$ ,  $\mathcal{X}_f = \frac{N-X_f}{2}$ ,  $\mathcal{Y}_f = \frac{M-Y_f}{2}$ ,  
 $\mathcal{X}_o = \frac{N-1}{2}$ ,  $\mathcal{Y}_o = \frac{M-1}{2}$ , としてオフセットを定義する。
  - (2) ネットワーク直径の最も短いノードに中心座標  $(\lfloor \frac{N}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$  を割り当てる。
  - (3) 近隣に座標を持たないノードを少なくとも1つ以上持ち、かつ座標が割り当てられているノードを選び、その近隣ノードの座標を  $HS$  を局所的に最小化するように割り当てる。
  - (4) 全てのセンサーノードが座標を取得するまで(3)を繰り返す。
- (1)では、オフセットは各層のユニットが中心に来るように選択する。(2)(3)(4)では、中心ノードから座標が導かれる。

### 4.3 順伝播処理

説明の簡略化のため、WSN内の  $N \times M$  個のセンサーノードが  $N \times M$  のアドレス空間に割り当てられたとする。つまり、センサーノードとユニットが1:1に割り当てられているものとする。この節では、各センサーノードの順伝播処理について説明する。ノードが欠損している場合については5章で説明する。

順伝播処理では、センサー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、各センサーノード  $(x, y)$  が以下の処理を行う。(1)入力層  $(x, y)$  のユニットを実行、(2)畳み込み層  $c_k^{(t)}(x + \mathcal{X}_c^{(t)}, y + \mathcal{Y}_c^{(t)})$  のユニットを実行、(3)プーリング層があるならば、 $p^{(t)}(x + \mathcal{X}_p^{(t)}, y + \mathcal{Y}_p^{(t)})$  のユニットを実行、(4) $(x, y)$  が出力層のユニットに割り当てられているならば出力層のユニットを実行。言い換えれば、座標  $(x, y)$  に対応するユニットは、センサーノード  $(x, y)$  で実行されることになる。

畳み込み層のユニット  $c_k^{(t)}(x + \mathcal{X}_c^{(t)}, y + \mathcal{Y}_c^{(t)})$  を実行するために、センサーノードは、前の層のプーリング層のユニット  $p^{(t-1)}(x + \mathcal{X}_p^{(t-1)} + i, y + \mathcal{Y}_p^{(t-1)} + j)$  から入力データを取得する  $(-\lfloor \frac{h_c^{(t)}}{2} \rfloor \leq i, j \leq \lfloor \frac{h_c^{(t)}}{2} \rfloor)$ 。そのため、前の層のプーリング層のユニットに対応するセンサーノード  $(x + \mathcal{X}_p^{(t-1)} + i, y + \mathcal{Y}_p^{(t-1)} + j)$  と通信する。そして以下の式によって畳み込み処理が行われる、 $f$  はReLUのような活性化関数であり、重みおよびバイアスをそれぞれ  $w$  および  $b$  とする。

$$c_k^{(t)}(x + \mathcal{X}_c^{(t)}, y + \mathcal{Y}_c^{(t)}) = f\left(\sum_{\lfloor -\frac{h_c^{(t)}}{2} \rfloor \leq i, j \leq \lfloor \frac{h_c^{(t)}}{2} \rfloor} w_{ij} \cdot p_{ij} + b\right) \quad (3)$$

ここで、 $p_{ij} = p^{(t-1)}(x + \mathcal{X}_p^{(t-1)} + i, y + \mathcal{Y}_p^{(t-1)} + j)$  である。もし前の層のプーリング層  $p^{(t-1)}$  がない場合には、前の層の

畳み込み層のユニット  $c_k^{(t-1)}(x + \mathcal{X}_c^{(t-1)} + i, y + \mathcal{Y}_c^{(t-1)} + j)$  から入力データを取得する。全結合層も同様にして処理が行われる。

プーリング層ではMAXプーリングを用い、 $p^{(t)}$  のユニットは以下の式によって処理が行われる。

$$p^{(t)}(x + \mathcal{X}_p^{(t)}, y + \mathcal{Y}_p^{(t)}) = \max_{1 \leq i, j \leq \lfloor \frac{s_p^{(t)}}{2} \rfloor} c_k^{(t)}(x + \mathcal{X}_c^{(t-1)} + i, y + \mathcal{Y}_c^{(t-1)} + j) \quad (4)$$

### 4.3.1 逆伝播処理

逆伝播処理における微分の計算は、CNNに沿って順伝播と逆方向に実行されるプロセスである。しかし、通常の逆伝播アルゴリズムでは、逆伝播によって求められた各層の全てのユニットの微分結果を用いて、その層におけるユニットの重みを更新するという集中型の処理である。つまり、この重み更新を行うためには全てのセンサーノードと通信を行う必要がある。これを避けるために、ユニットが後の層のユニットから微分結果を取得し、その結果のみを用いて重みを更新する分散アルゴリズムを設計する。したがって、提案手法では、各ユニットは他のユニットと重みを共有しないものとする。

出力層のユニットが実行され、正解データを取得すると、微分  $\delta$  を計算し、その結果を前の層のユニットへ送信する。微分  $\delta$  を受け取ったユニットは、 $\delta$  および入力データを用いて、パラメータ勾配および前の層のユニットに送信するための  $\delta$  を計算する。このとき、パラメータ勾配を用いて重みの更新も行う。

ここで出力層の  $\delta$  は、出力結果を  $z$ 、正解データを  $t$  とすると、以下の式によって求められる。

$$\delta = z - t \quad (5)$$

後の層から  $\delta$  を受け取った後、前の層へ送信するための  $\delta$  は、以下の式によって求められる。このとき、 $w_i^{(t)}$  および  $\delta_i^{(t)}$  は、 $T-1$  層での  $i$  番目のユニットから  $T$  層での  $j$  番目のユニットへの重みおよび微分を表すとする。

$$\delta_i^{(t)} = \sum_j \delta_j^{(t+1)} (w_j^{(t+1)} f'(u_j^{(t)})) \quad (6)$$

また、以下の式によってパラメータ更新を行う。

$$w_i^{(t+1)} = w_i^{(t)} - \epsilon \delta_i^{(t)} z_i^{(t)} \quad (7)$$

$\epsilon$  は学習率で  $z^{(t)}$  は前の層の出力結果を表す。

## 5. ノード欠損時の処理

上記の設計は、 $N \times M$  の座標全てにセンサー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、実環境を考慮すると、必ずしも全ての座標にセンサーノードが割り当てられているとは限らない。例えば、センサーノードが11個しか配置されていない場合、 $4 \times 3$  の空間を網羅

することはできない。また、センサーノードの動作が必ず保証されているわけではなく、センサーノードが故障する場合も考えられる。このようにセンサーノードが欠損している場合でも学習処理を可能にするために、センサーノード近辺の代替処理を定義する。

センサーノードがセンサーデータを生成するため、センサーノードの欠損は、CNNが入力データの1つを失うことを意味する。そこで、欠損部分の入力データを必要とするユニットは、その部分を0で補間する。この補間による入力データは、畳み込み処理の線形結合性より無視される。また、センサーノードの欠損は入力データの欠損だけでなく、畳み込み層の出力結果の1つを失うことでもある。そのため、畳み込み層の出力を必要とするユニットは、同様に0として補間する。この補間データは、次の層の畳み込み層およびプーリング層の処理によって無視される。次の層が畳み込み層の場合は、上述のように線形結合性により無視され、プーリング層の場合は、活性化関数 ReLU により入力データが非負の値であり、それに MAX プーリング処理を行うため、0は無視される。プーリング層におけるユニットの欠損も同様である。

全結合層および出力層の場合は、処理を1つのノードに集約して行うため、このセンサーノードの欠損は、順伝播および逆伝播処理を行うことができないということである。したがって、この場合には、代替ノードを設け、そのノードが処理を行うものとする。

## 6. 温度センシングによる異常検知と評価

### 6.1 学習に使用したデータセット

まず、深層学習の実行にあたり用いたデータセットについて述べる。大阪府内の  $1,400m^2$  超の実ラウンジスペースに設置した50個の温度センサーが取得したデータを学習のためのデータセットとして使用した。ラウンジの空間を  $17 \times 25$  のメッシュ状の小領域に分割し、各小領域に取得された室温をマッピングした画像状のデータの集合として構成する。温度センサーとして、工場の温度管理などで使用される信頼性の高い無線温度センサー RTR-503 を用いた。また設置場所に制約があり、 $17 \times 25$  の全てにセンサーを準備することができなかったため、センサーの存在しない位置の温度データは実際に取得できた位置の温度データを用いて補間し、面的な温度分布データを作成した。補間の方法として、50個の温度センサーから取得した実温度データに対し、補間位置から実温度データまでのマンハッタン距離の逆数を重みとして与え、全ての実温度データに対して重み平均を取ることで、温度データの補間を行った。一例として、ある時刻における実際に取得できた計測値のみから構成した室温マップと、補間処理によって作成した室温マップをそれぞれ図4(a)と図4(b)に示す。本研究では2016年8月26日から10月27日までの間に30分間隔で

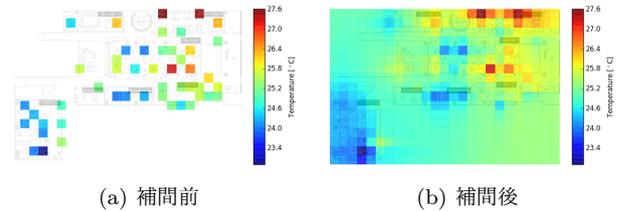


図4 ラウンジの温度データ一例：(a) 補間前 (b) 補間後

記録された2,961個の室温マップを深層学習に使用した。

このような二次元の温度データに対し、本研究では最終的に人の快・不快、機器の故障といった温度分布からでは判別が困難な対象を深層学習により識別することを目的としている。しかしながら、現在我々のデータセットでは温度情報のみが得られており、その空間での通常/異常状態を示す正解ラベルは付与されていない。したがって、本評価では、太陽光や壁、空調設備からの距離などにより周囲と比較して温度差が大きいスポットが発生したときに居住者が不快感を持つと仮定し、それを異常状態と定義したうえで、異なるハイパーパラメータ設定において異常状態を正しく検知できるかどうかを評価する。本研究では、収集した2,961セットの室温マップに対し、室温の不均衡が人の快適度に影響を及ぼすという想定の下、 $5 \times 5$ の部分空間内の温度データの分散が1.0以上、すなわち局所的に温度分布の偏りが存在する場合は異常、そうでない場合は正常とした正解ラベルを機械的に付与した。この設定のもとでは、2,961個のデータセットに対し、およそ17.46%のデータが異常として判定される。

### 6.2 分散学習によるデータ通信量の変化

まず、データ通信量という点において、ハイパーパラメータを変更した時にデータ通信量がどのように変化するかを示す。これにあたり用いたネットワークは、 $25 \times 17$ の入力層、畳み込み2層、プーリング1層、全結合層1層、出力層1層からなるネットワークである。学習回数および畳み込みのフィルタ数は、データ通信量の増加率に影響しないため一定とする。データ通信量の変化をみるために、1台のPCに $25 \times 17$ の仮想ノードをpythonで用意した。そして、これらのセンサーを仮想環境内で分散学習をし、シミュレーションを行った。

その結果、フィルタサイズおよびプーリングサイズを変更した時のデータ通信量の変化はそれぞれ以下のグラフ図5のようになった。このことから、フィルタサイズおよびプーリングサイズを大きくすると、データ交換先のノード数が増加するため、データ通信量は増加することが分かった。

次に、データ通信量という点において、ニューラルネットワークを変更したときにデータ通信量がどのように変更するかを示す。図6に隠れ層  $T$  の数を変化させたときのデータ通信量の変化を示す。この結果から、畳み込みやプーリングによってニューラルネットワークが深くなるほ

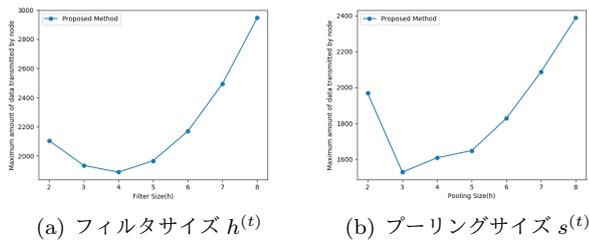


図 5 フィルタサイズ  $h^{(t)}$  およびプーリングサイズ  $s^{(t)}$  を変更したときの最大データ通信量の変化

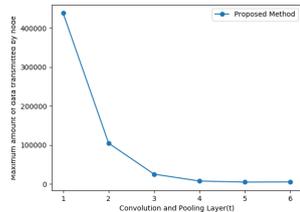


図 6 隠れ層  $T$  を変更したときの最大データ通信量の変化

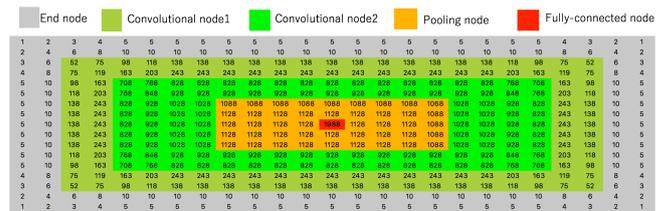
ど、出力層へのデータサイズが小さくなるため、データ通信量は減少することが分かる。

### 6.3 学習精度の評価

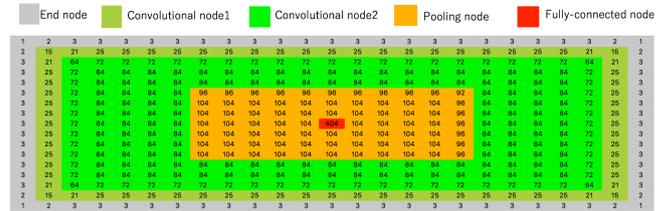
提案手法の課題として、データ通信量と学習精度をできるだけ最適にするような CNN のハイパーパラメータを選択する必要があるということが挙げられる、また、提案手法では、通常の CNN と異なり、畳み込みのフィルタのパラメータ更新をユニット毎に行うため、それによる学習精度の低下が懸念される。そこで通常の CNN との学習精度の比較を行った。

学習精度の評価を行うにあたり、データセットのうち 9 割をトレーニングデータ、1 割をテストデータとした。そしてハイパーパラメータはそれぞれ下記のように設定した。提案手法による分散学習の場合、フィルタ数： $K^{(1)} = K^{(2)} = 2$ 、フィルタサイズ： $h^{(1)} = h^{(2)} = 3$ 、プーリングサイズ： $s^{(2)} = 2$ 、パディングサイズ：0、全結合層のユニット数：150、バッチサイズ：3、学習回数：7 とした。これに対し、通常の CNN の場合、フィルタ数： $K^{(1)} = K^{(2)} = 10$ 、フィルタサイズ： $h^{(1)} = h^{(2)} = 5$ 、プーリングサイズ： $s^{(2)} = 2$ 、パディングサイズ：0、全結合層のユニット数：300、バッチサイズ：3、学習回数：10 とし、学習精度が最適となるよう設定した。また、学習精度を比較する際、訓練データ数が比較的少ないため、学習開始時にランダムに設定する各層の重みの初期値によって学習精度は計算ごとに異なる。そこで、各条件で 10 回ずつ学習と評価を行った結果を平均化したものを学習精度とした。

その結果、提案手法は 95.570%、通常の CNN は 97.095% となった。分散学習において、データを中心に集約し、最適なハイパーパラメータの選択を行った場合、通信量は図 7(a) となる。一方、提案手法におけるハイパーパラメータの選択を行った場合、通信量は図 7(b) となり、赤色で示される



(a) 最適なパラメータ選択の場合



(b) 制約の中でのパラメータ選択の場合

図 7 各センサーノードのデータ通信量 (温度データ)

最大通信量は、かなりの減少が見られることが分かった、

次に、提案手法と通常の CNN の学習において、同じハイパーパラメータの選択をした場合の学習精度の比較を行った。その結果、提案手法では 95.570%、通常の CNN では 95.6% となり、ほぼ同じ結果となった。このことから、提案手法における、重み共有を行わない畳み込みフィルタのパラメータ更新でも、通常の CNN と同程度の学習精度を達成可能であることが分かる。

### 6.4 センサー NW のノード欠損による影響

5章で述べたように、実環境を考慮すると、センサーノードの故障に対処できるようにすることは重要な課題となる。センサーノードの故障によるノードの欠損は、CNN のユニットが欠損した不完全な状態での学習となる、この状態での学習を定量化するために、 $25 \times 17$  個のセンサーノードから意図的に一部のセンサーノードを欠如させる。ノードが欠損した場合のユニットの出力としては、5章で述べた方法を用いる。

$25 \times 17$  個のセンサーノードのうち、ランダムに 1 割 (42 個) のノードを欠如させた場合、学習精度は 94.9% となった。同様にランダムに 2 割のノードを欠如させた場合の学習精度は 94.4% であった。この結果から、センサーノードの欠損によって、大幅に学習精度が低下することはないといえる。これはセンサーノードの入力の欠損は、学習精度にそれほど影響をあたえるものではなく、欠損した入力データが、特徴量の決定的な部分ではない限り、たとえ、入力データが欠損したとしても、周囲のユニットによってその欠損が補われるのではないかと考えられる。

### 6.5 HopStretch の評価

4.2 章で  $HS$  というオーバーレイネットワークと物理ネットワークのリンカー一致性を示すメトリクスについて説

明した。これについて、様々な WSN における  $HS$  の評価を行う。ここで、WSN として、(1) 直交されたセンサーノード間のみが通信接続されたトポロジ、および (2) 直交に加え、斜めのセンサーノード間が通信接続されたトポロジ、の 2 つを用いた。この場合における、 $HS$  はそれぞれ (1)1.80, (2)1.22 となり、ネットワークの接続密度が高い WSN ほど、 $HS$  の値は 1 に近づくことが分かる。

## 7. おわりに

本研究では、無線センサーネットワーク内で深層学習を実行するプロトコルを提案した。提案手法では、CNN のユニットを無線センサーノードに割り当て、センサーノード間の通信によって CNN の学習を分散実行する。1400 $m^2$  超の実ラウンジスペースでの温度データセットを用いた性能評価実験を行った結果、提案手法は、通常の CNN と同程度の学習精度を維持しながら、十分妥当な通信量のもとで、WSN 内で CNN を分散実行可能であることを確認した。また、一部のノードが故障した場合にも、学習精度に大きな影響を与えることなく深層学習を分散実行することが可能であることも示した。

提案手法による分散学習を実環境に適用する場合、通信品質やエネルギー消費、プロセス間の同期などの影響についても考慮する必要がある。したがって、今後の計画としては、小型のエネルギーハーベスティングなセンサーノードに分散学習を実装し、それらの影響をを測定していくことがあげられる。最終的には、この分散学習システムをエネルギーハーベスティングなセンサーに導入し、WSN 内でエネルギー効率の高い深層学習を実現していきたいと考えている。

## 謝辞

本研究成果の一部は、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」ならびに JSPS 科研費 15K12019 の助成を受けたものです。

## 参考文献

- [1] Scherer, D., Müller, A. and Behnke, S.: Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition, *Proceedings of the 20th International Conference on Artificial Neural Networks: Part III, ICANN'10*, Berlin, Heidelberg, Springer-Verlag, pp. 92–101 (online), available from <http://dl.acm.org/citation.cfm?id=1886436.1886447> (2010).
- [2] Babenko, A., Slesarev, A., Chigorin, A. and Lempit-sky, V.: Neural codes for image retrieval, *Proceedings of European conference on computer vision*, Springer, pp. 584–599 (2014).
- [3] : [http://deeplearning.stanford.edu/wiki/index.php/Neural\\_Networks](http://deeplearning.stanford.edu/wiki/index.php/Neural_Networks).

- [4] Tu, J. V.: Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes, *Journal of clinical epidemiology*, Vol. 49, No. 11, pp. 1225–1231 (1996).
- [5] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T.: DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition., *Icml*, Vol. 32, pp. 647–655 (2014).
- [6] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks, *arXiv preprint arXiv:1312.6229* (2013).
- [7] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T.: Caffe: Convolutional architecture for fast feature embedding, *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, pp. 675–678 (2014).
- [8] Buyya, R., Calheiros, R. N. and Dastjerdi, A. V.: *Big Data: Principles and Paradigms*, Morgan Kaufmann (2016).
- [9] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., Le, Q. V. et al.: Large scale distributed deep networks, *Advances in neural information processing systems*, pp. 1223–1231 (2012).
- [10] Schmidhuber, J.: Deep learning in neural networks: An overview, *Neural networks*, Vol. 61, pp. 85–117 (2015).
- [11] 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).
- [12] 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).
- [13] 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).
- [14] Rabbani, A. and Keshav, S.: The SPOT\* Personal Thermal Comfort System, *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, BuildSys '16, New York, NY, USA, ACM, pp. 75–84 (online), DOI: 10.1145/2993422.2993578 (2016).
- [15] Chiguchi, M., Yamaguchi, H., Higashino, T. and Shimoda, Y.: Human thermal comfort estimation in indoor space by crowd sensing, *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 45–50 (online), DOI: 10.1109/SmartGridComm.2016.7778736 (2016).
- [16] Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C.: *Introduction to Algorithms, Third Edition*, The MIT Press (2009).
- [17] hua Chu, Y., Rao, S. G., Seshan, S. and Zhang, H.: A case for end system multicast, *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 8, pp. 1456–1471 (online), DOI: 10.1109/JSAC.2002.803066 (2002).