Vol.2017-DPS-170 No.18
Vol.2017-CSEC-76 No.18
2017/3/2

無線センサーネットワークにおける深層学習の分散実行に関する検討

福島 悠太[1,a)]　三浦 太樹[1,b)]　濱谷 尚志[1,c)]　山口 弘純[1,d)]　東野 輝夫[1,e)]

概要：近年の IoT データのクラウド集約と機械学習は，通信にかかる電力やクラウドのデータ処理量など様々な制約により限界を迎えようとしており，処理集中を抑制しながら効率よくセンシングおよび学習解析処理を行えるアーキテクチャが望まれる．本研究では深層学習を対象に，それをデータ発生源であるセンサー機器からなる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に取得するデータ（例えば温度分布など）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当てる．それらユニットが用いる畳み込み処理（フィルター）やプーリング，および逆伝搬のデータ処理範囲を制約することで，各ノードへの処理量をほぼ均等に分散しながら近隣ノードとのデータ交換を必要最小限に抑制する．提案手法の有効性を評価するため，センサーネットワーク内の特定ノードがデータ集約と学習を行う集約型とのノード毎のデータ通信量の比較を行った結果，集約型の特定ノードの負荷と比較して各ノードの通信量を抑制可能な深層学習パラメータ設計が可能であることがわかった．また，実在するラウンジスペースにおいて収集した 50 地点の室温データの異常判定を行うシナリオにおいて，上述のパラメータ設定を用いても十分な精度の判定器を構成できることがわかった．

A Study on Decentralization of Deep Learning in Wireless Sensor Network

Yuta Fukushima[1,a)]　Daiki Miura[1,b)]　Takashi hamatani[1,c)]　Hirozumi Yamaguchi[1,d)]
Teruo Higashino[1,e)]

1. はじめに

現在の IoT データ処理は，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たクラウドヘビー型である．一方で，Google TensorFlow などの通常の PC アーキテクチャで実行可能な深層学習パッケージも増えており，データをクラウドではなくサイト毎のローカルゲートウェイ等で集約して学習・解析処理を行うことも現実的になってきている．しかし，それらの集約型アーキテクチャはいずれも集約ノードへのデータトラフィック集中および学習データ処理負荷の増大が課題となる．特に IoT センサーネットワークでは多数のセンサーノードからのデータ集約にかかるトラフィック負荷は無視できず，集約ノードは学習機能を備えた高性能ノードである必要があり，可用性の点でも望ましくない．

一方で，現状の IoT 機器はインターネット接続機能を有するデータ生成デバイスであるが，将来的に大量に展開される超小型 IoT 機器一つひとつの処理性能やメモリ量が向上していくと想定される．したがって，これらの IoT 機器の処理性能や搭載センサをシームレスに連携・融合させ，

[1] 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

センサーノード全体でセンシングデータを分散学習処理することで，センシング，学習，フィードバックといった一連のデータ活用がセンサーネットワーク内で可能になり，センサーネットワーク自体が知的なデータ処理基盤となり得る．

そこで，本研究では深層学習によるデータ解析処理を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手法を提案する．提案手法ではメッシュ型の無線センサーネットワークが面的に取得するデータ（例えば温度分布など）を対象とし，近年物体認識および画像検索などに多数用いられる畳み込みニューラルネットワーク（CNN）[1,2] におけるユニットの役割を各センサーノードに割当てる．それらユニットが用いる畳み込み処理（フィルター）やプーリング，および逆伝搬のデータ処理範囲を制約することで，各ノードへの処理量をほぼ均等に分散しながら近隣ノードとのデータ交換を必要最小限に抑制する．

大量のデータを扱う深層学習では，並列分散実行による処理効率化を図る研究も多く，GPU を用いた並列化手法 [3] や，SINGA と呼ばれる学習システムを用いた手法 [4] などが知られている．これらの手法は，深層学習における計算処理の並列化およびニューラルネットワークの分割によって，計算コストの効率化を図る手法であるが，提案手法は，センサーネットワークを構成する個々のノードに深層学習の個々のユニットの役割を対応付けることで深層学習そのものをセンサーネットワーク全体で分散実行することを目指している．センサーネットワークのスケールを活用し，個々のノードの処理負荷を均等に分散させ，また深層学習のパラメータ設定も工夫することで，ノード間の通信量も抑制する工夫をしている．最終的には各ノードがエネジーハーベスティングに近い電力で処理を実行することで全体の学習が省電力に実現されることを目指している．

提案手法の有効性を評価するために，ノード毎のデータ通信量を，全入力データをセンサーネットワーク内の特定ノードに集約する方法（集約型）と比較した．ノードのデータ通信量は，深層学習のニューラルネットワークモデルおよびフィルタサイズ等のハイパーパラメータに依存するため，深層学習のニューラルネットワークモデルを固定し，ハイパーパラメータを変更したうえで評価した．その結果，データ集約ノードにおける負荷を軽減しながら，通信量削減を実現するハイパーパラメータが存在することが確認できた．また，実在する 1000m^2 超の屋内ラウンジスペース内の 50 点で収集したおよそ 60 日間の気温データに対して温度の異常検知を行うシナリオにおいて，前述のニューラルネットワークモデルを用いて，ハイパーパラメータを変更したときの学習精度を評価した．その結果，ハイパーパラメータ制約による学習精度への影響は微小であることが確認できた．すなわち，提案手法では学習精度を損なうことなく負荷およびトラフィックを均等分散しながら深層学習を自律分散的に実行できることが確認できている．

2. 関連研究

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きた．音声認識，物体認識，画像検索，および自然言語処理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．一般的に，深層学習はデータ量が増加すればするほど高い精度を得ることができるが，画像などの高精細データにおける訓練では数千万のパラメータと数十億の訓練データが必要となる．したがって，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てきている．以下ではそれらについて述べるとともに，本研究の位置づけを述べる．

2.1 GPU を用いた深層学習の並列化

文献 [3] 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．GPU は数千の ALU コアを搭載しているため数値演算能力が優れる一方，メモリ制限が課題となる．この問題を解決するため，マルチ GPU を用いたデータ並列化，モデル並列化，およびデータモデル並列化からなる分散システムを導入している．データ並列化は，各 GPU が同じモデルと異なるデータセットで学習を実行し，その後異なる GPU から学習したパラメータ勾配を同期する手法である．モデル並列化は，大規模なモデルを分割し，分割したモデルを各 GPU で担当し，各 GPU が同じデータセットを異なるモデルで学習する手法である．しかしデータ並列化では，計算ノードが多数ある場合にはスムーズに学習を行うため学習率を下げる必要があり，モデル並列化では，モデル間の通信コストが問題となる．これに対し，データモデル並列化は，全結合層と畳み込み層の特性から，畳み込み層をデータ並列化し，全結合層をモデル並列化することで，大規模な深層学習のモデルのより高速な学習を実現している．他に深層学習の並列化として，文献 [5] では 16000 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 [6] では，6000 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．

2.2 深層学習の分散型アーキテクチャ

文献 [4] では，大規模な深層学習のモデルを学習するための手法として，SINGA という一般的な分散型の深層学習のプラットフォームを提案している．SINGA は畳み込みニューラルネットワーク（CNN），ボルツマンマシン（RBM），および再帰ニューラルネットワーク（RNN）の一般的な深層学習モデルをサポートする．SINGA は，パ

ラメータ勾配を計算する TrainOneBatch と順伝搬を行う NeuralNet からなる Worker ユニットと，結果を集約する Stub ユニット，パラメータを更新する Server ユニットの 3 つのコンポーネントを提供し，これらのユニットをユーザーが設定することで分散深層学習を行える．Worker の NeuralNet では CNN や RBN，RNN のニューラルネットワークを設定でき，大規模なモデルでは，次の 4 つの並列化手法（(1) 全てのレイヤーを異なるサブセットに分割，(2) 1 つのレイヤーをバッチ次元によってサブレイヤーに分割，(3) 1 つのレイヤーを特徴次元によってサブレイヤーに分割，(4) (1)〜(3) の手法の組み合わせ）が提供される．また，TrainOneBatch では，パラメータ勾配を計算するために順伝搬型ニューラルネットワークや RNN などで用いられるバックプロパゲーションとエネルギーモデルで用いられる対比分散アルゴリズムが提供されている．

SINGA では Worker と Server を用いた多様な同期および非同期のフレームワークを提供でき，例えば Sandblaster [5] といった同期フレームワークや Downpour [5] や Distributed Hogwild [7] といった非同期フレームワークが知られている．同期フレームワークは，分散により 1 回の学習速度を早くすることができ，非同期フレームワークは収束率を高めることができる．学習速度と収束率はトレードオフの関係であり，異なるフレームワークを組み合わせて収束率と学習速度における最適性を得ることができる．

2.3 本研究の位置づけ

前述のように，深層学習の分散実行は従来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，提案手法は面的なセンサーメッシュネットワークから得られるデータを近隣ノードと交換することで，深層学習に必要な畳み込みやプーリング（順伝搬プロセス）が実行され，それに応じて逆伝搬プロセスも実行されることでデータセンシングを行うセンサーノード全体で深層学習を実現する新しい分散実行プロトコルの設計手法である．センサーネットワークのスケールを活用し，個々のノードの処理負荷を均等に分散させ，また深層学習のパラメータ設定も工夫することで，ノード間の通信量も抑制する工夫をしており，これまでにない独創的な取り組みであるといえる．

3. 提案手法

3.1 概要

本手法では，図 1 のように，オフィスや工場，フィールドなどの空間に配置された多数の IoT センサーノード（以下，ノードとよぶ）が無線メッシュネットワークにより接続された環境を想定する．メッシュ状に配置されたノードから対象空間内の面的なデータ（例えば人の分布や温度分布，土壌水分分布など）を取得し，そのデータに対し空間全体で異常検知などの識別を行うことを目的とする．本手法では無線メッシュネットワークが深層学習における中間層と類似の構造を持っていることに着想を得て，各ノードが生成するデータを近隣のノードと交換し，各センサーノード自身が深層学習における畳み込み・プーリング処理を行うプロトコルを提案する．

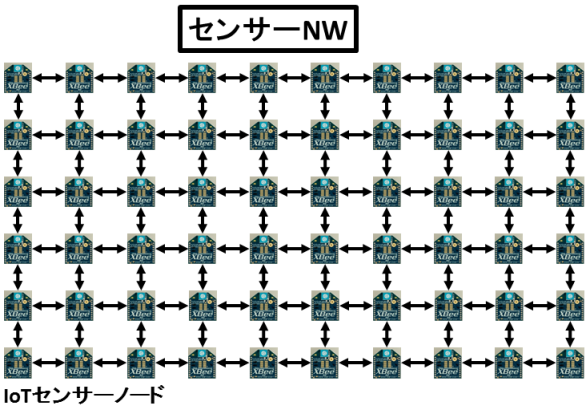


図 1 想定環境

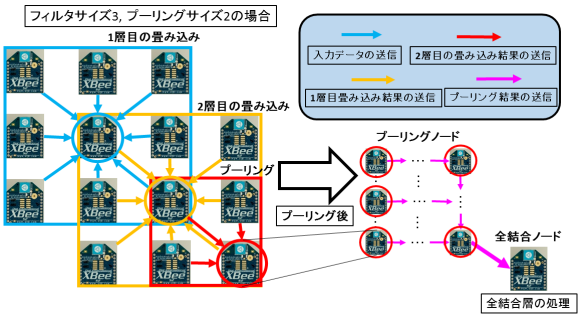


図 2 順伝搬処理

3.2 アーキテクチャ

一般に各ノードはメモリやプロセッサ，電力の制約から大量のデータ処理は困難であるものの，畳み込み計算など単純な部分計算処理は十分可能である．また電力の制約から隣接ノードとのみ直接無線通信が可能であるとし，その他のノードとはマルチホップ通信を行うとする．本手法では，深層学習におけるニューラルネットワークを構成する入力層，中間層（畳み込み層・プーリング層），全結合層における順伝搬（推定），逆伝搬（学習）処理について，それぞれ以下のようにノードに分散する．

3.2.1 順伝搬処理

図 2 に畳込み 2 層とプーリング 1 層の隠れユニットを持つ場合の順伝搬の分散実行を示す．

以下，順伝搬の各ユニットで行われる畳込みに必要なデータを有するノードを周辺ノードとよぶ．簡単のため，すべてのノードはセンサーデータを生成するものとし，入力層のユニットの役割を果たすものとする．また，周辺

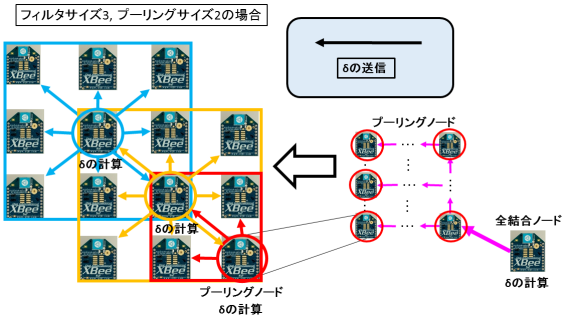


図 3　逆伝搬処理

ノードを有するすべてのノードはそれぞれそのデータを用いた畳込みを行うため，畳込み層のユニットの役割を果たすものとする．以下では畳込み層数を K，プーリング層が全結合層の直前に 1 つ挿入されている CNN を対象とする．

(1) 入力層：入力層の役割を果たす各ノードは，取得したデータを周辺ノードに（シングルホップまたはマルチホップ通信で）送信し，各ノードが畳込み処理に必要なデータ（フィルタサイズに相当するデータ）を伝搬する．

(2) k 番目の畳込み層：畳込み層の役割を果たす各ノードは，$k = 1$ のときはステップ (1) からの，$k > 1$ のときは $k - 1$ のときのステップ (2) からのデータから畳込み計算を行い，その結果を周辺ノードに送信する．$k < K$ のときは，$k = k + 1$ として (2) を繰り返す．

(3) プーリング層：プーリング層の役割を果たす各ノードは，$k = K$ のときのステップ (2) からのデータからプーリングに必要なデータを取得し，プーリング計算を行う．

(4) 全結合層：全結合層ではステップ (3) の全プーリング層ノードの情報を用いて計算を行う．全結合層をいずれかのノードに割当て，そのノード（以降，全結合ノードとよぶ）が推定結果の出力を行う．

3.2.2　逆伝搬処理

図 3 に逆伝搬の分散実行を示す．

収集したデータに対し教師データが与えられる場合，以下のように誤差逆伝搬法によるニューラルネットワークのパラメータ更新を行う．

(1) 全結合層：与えられた教師データと予測結果を基に，パラメータ更新に必要な入力に関する微分 δ を計算する．得られた微分結果は前の層のパラメータ更新に必要であるため，マルチホップ通信を用いて直前の層へ δ を送信する．

(2) プーリング層・畳込み層：各ノードは，次の層から受け取った δ，および前の層の出力結果を用いてパラメータ勾配を計算しパラメータの更新を行う．さらに前に層が存在する場合には δ を逆伝搬する．

提案手法では，全結合層では直前の層の全ユニットを参照することが可能である一方で，畳込み層，プーリング層はセンサーネットワーク上の各ノードに実装されるため，近隣のノード以外は参照しない方針で実現する．これはすなわち，誤差逆伝搬法において各ノードが持つ重みパラメータを全体で共有することができないことを意味する．したがって，提案手法では各ノードは重みの共有を行わず，個別にフィルタの更新を行う分散実行方針を採用する．

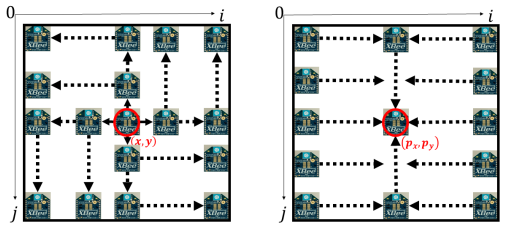
4.　分散実行プロトコルの設計

本章では，3 章で与えた深層学習の分散実行方針を実現するプロトコルの設計について述べる．以下，入力データは $m \times n$ の 2 次元データとし，畳込み層のフィルタサイズを h，フィルタ数を k，プーリングサイズを s で表す．

4.1　ノード動作の設計

前述のように，ノードの役割は，データの送受信，畳み込み処理，プーリング処理，パラメータ更新に必要な δ の計算およびパラメータの更新である．以降，センサー位置を左上を原点とした XY 正整数座標系で表現するものとし，座標 (x, y) のノードを I_{ij} とする．I_{ij} が保持すべきデータは各畳み込み層・プーリング層における重みとバイアス，パラメータ更新用の微分 δ，および近隣のノードから受け取ったセンサーデータ，畳み込み結果，プーリング結果である．以降でノード I_{ij} が行うべき処理を述べる．

- センサーデータ交換
  - 畳込み層実行のためのデータ交換：図 4(a) にデータの流れを示す．まず $I_{i,j}$ は $I_{i-1,j}$，$I_{i,j-1}$，$I_{i+1,j}$，$I_{i,j+1}$ の 4 つの隣接ノードに入力データを送信する．また，$I_{x,y}$ が生成した入力データ（$I_{x,y}$ の入力データとよぶ）を受信し，かつ $i - \frac{h}{2} < x < i + \frac{h}{2}$，かつ $j - \frac{h}{2} < y < j + \frac{h}{2}$) である場合，以下にしたがって転送する．(i) $I_{x,y}$ の入力データを $I_{i,j-1}$ から受信した場合，$I_{i,j+1}$ に送信する．$i = x$ のとき，$I_{i+1,j}$ にも送信する．(ii) $I_{x,y}$ の入力データを $I_{i-1,j}$ から受信した場合，$I_{i+1,j}$ に送信する．$j = y$ のとき，$I_{i,j-1}$ にも送信する．(iii) $I_{x,y}$ の入力データを $I_{i,j+1}$ から受信した場合，$I_{i,j-1}$ に送信する．$i = x$ のとき，$I_{i-1,j}$ にも送信する．(iv) $I_{x,y}$ の入力データを $I_{i+1,j}$ から受信した場合，$I_{i-1,j}$ に送信する．$j = y$ のとき，$I_{i,j+1}$ にも送信する．これにより，フィルタサイズ h 内のノードに入力データが送信される．
  - プーリングおよび全結合処理のためのデータ交換：図 4(b) にデータの流れを示す．以下では，ノード $I_{i,j}$ がプーリングを行うノード（以下，プーリングノードとよぶ）に対するデータ送信，あるいはプーリングノードが全結合ノードに対し行うデータ送信について述べ，それらの送信先ノードを I_{p_x,p_y} で表す．全結合ノードの座標は既知とするが，プーリングノードの座標 (p_x,p_y) はニューラルネットワークに依存し，



(a) 畳み込み　(b) プーリングおよび全結合

図 4　センサーデータ交換

フィルタサイズ h，プーリングサイズ s およびデータ送信を行うノードの座標 (i, j) で求められる．$i = p_x$ かつ $j = p_y$ のとき，データ送信を行うノード自身がプーリングノードであるため，データ送信は行わない．また，それ以外のプーリングを行わないノードは，$i = p_x$ のとき，$j > p_y$ ならば $I_{i,j-1}$ に，$j < p_y$ ならば $I_{i,j+1}$ に入力データおよび受信データを送信する．$i \neq p_x$ のとき，$i > p_x$ ならば $I_{i-1,j}$ に，$i < p_x$ ならば，$I_{i+1,j}$ に入力データおよび受信データを送信する．

– パラメータ更新のためのデータ交換
パラメータ更新の微分 δ を送信する際は，δ を送信する送信先のノードの座標が，δ を計算したノードにより既知であるとする．順伝搬処理の際，$I_{x,y}$ のデータをどのノードから受信したを記憶しておくものとし，そのノードを I_{x_i,y_j} とする．δ の送信先のノードが $I_{x,y}$ であるとき，δ を I_{x_i,y_j} に送信する，

- 畳込み処理
h^2 個の入力データを取得したノード $I_{i,j}$ は以下の式によって畳み込み処理を行う．入力データを x_{ij}，フィルタの重みを w_{ij}，バイアスを b，活性化関数を f，畳み込み結果を z_{ij} とする．

$$z_{ij} = f(\sum_{j-\frac{h}{2}}^{j+\frac{h}{2}} \sum_{i-\frac{h}{2}}^{i+\frac{h}{2}} x_{ij} w_{ij} + b)$$

- プーリング処理
s^2 個の入力データを取得したプーリングノードは，以下の式によってプーリング処理 (MAX プーリング) を行う．入力データを x_{ij}，プーリング結果を z_{ij} とする．

$$z_{ij} = max(x_{i-\frac{s}{2},j-\frac{s}{2}}, \cdots, x_{i+\frac{s}{2},j+\frac{s}{2}})$$

- パラメータ更新
次層から δ を取得したノードは，以下の式によって前層に送信する δ を求める．重みを w_{ij}，次層を $l+1$，現在の層を l，活性化関数を用いる前の前層の出力結果を u_{ij} とする．

$$\delta_{ij}^{(l)} = \sum_{i,j \in (l+1)} \delta_{ij}^{(l+1)} (w_{ij}^{(l+1)} f'(u_{ij}^{(l)}))$$

また，以下の式によってパラメータ更新を行う．更新前のパラメータを $w^{(t)}$，更新後のパラメータを $w^{(t+1)}$，前の層の出力結果を z，学習係数を ϵ とする．

$$w^{(t+1)} = w^{(t)} - \epsilon \delta z$$

以上の処理を，データ交換，処理，データ交換，処理，...，データ交換（全結合ノードへ），データ交換，パラメータ更新，データ交換，パラメータ更新，... という流れで繰り返すことにより 1 回の学習を行う．

4.2　全結合ノードの実装

全結合ノードは，全結合層の順伝搬処理，誤差逆伝搬法における微分 δ の計算およびパラメータの更新を行う．全結合ノードが保持すべきデータは正解データ，全結合層における重みとバイアス，パラメータ更新用の微分 δ，プーリング結果，および全結合層の結果である．以降でそれぞれの処理の詳細を述べる．

- 全結合の順伝搬処理
入力データを x_{ij}，全結合層の重みを w_{ij}，バイアスを b，活性化関数を f，出力結果を z_{ij}，入力データ集合を P とすると，z_{ij} は以下で得られる．

$$z_{ij} = f(\sum_{i,j \in P} x_{ij} w_{ij} + b)$$

- 微分 δ の計算
出力層の活性化関数としてソフトマックス関数，誤差関数として交差エントロピーを用いるとする．また，出力結果を z_{ij}，正解データを t_{ij} とする．このときの出力層の微分 δ_{ij} は以下で得られる．

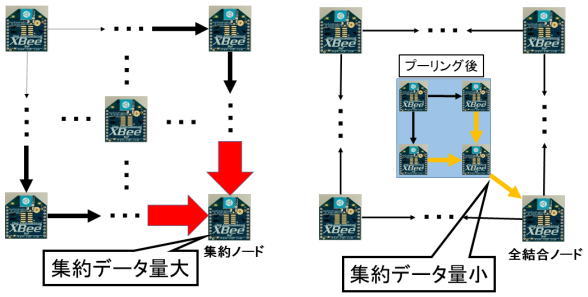
$$\delta_{ij} = z_{ij} - t_{ij}$$

全結合ノードは，プーリング結果を集約後，全ての全結合処理，および全結合層の数に応じた微分 δ の計算を実施する．

5.　通信データ量解析

本章では，3 章および 4 章で述べた提案手法において，全結合ノードに集約されるデータ量（集約データ量）の考察を行う．

従来の集約型では，図 5(a) のように，データを集約するノード付近のノードに集約データが集中する．一方，提案手法では，センサーネットワーク内の各ノードが近隣のノードとのローカルな通信により，データを各ノードに分散して処理を行うため，各ノードに対しほぼ均等にデータトラフィックを分散することが可能となる．更に，全ユニットを参照しなければならない全結合処理において，図 5(b) のように，全結合ノードにデータを集約する前に，センサーネットワーク内の各ノードで畳み込みおよびプーリング処理を行うことにより，全結合ノードに集約される



(a) 集約型　(b) 分散型

図 5　集約データ量

データ量を削減することが可能となる．

また，集約データ量は，従来の集約型の場合は入力データサイズのみに依存し，提案手法による分散型の場合は入力データサイズ，ニューラルネットワークおよびハイパーパラメータに依存する．そこで次節において，ニューラルネットワークを固定し，入力データおよびハイパーパラメータによる集約データ量の比較を行う．

5.1　単純なデータ集約プロトコルとの比較

提案手法の有効性を評価するため，まず，従来の集約型と本研究の提案手法による分散型のノード単位の集約データ量を比較する．集約データ量は 1 回の学習で集約するデータ量の合計とし，各ノードはマルチホップ通信によってデータの送信を行うものとする．提案手法を適用するニューラルネットワークの中間層は畳込み 2 層，プーリング 1 層，全結合層 2 層の計 5 層を想定する．センサーノードは $m \times n$ のメッシュ状に配置されており，入力データは $m \times n$ の 2 次元データであるとする．

この環境では，従来の集約型における集約データ量は，入力データ数 mn に等しい．すなわち，センサー数の増大に伴い，図 5 のように全結合ノード付近のノードの集約データ量が増大する．これに対し，提案手法のフィルタ数を k，フィルタサイズを h，プーリングサイズを s とすると，1 層目 (畳込み層) では，フィルタサイズ分のデータを集約するため，h^2-1 となる．2 層目 (畳込み層) では，1 層目で複数のフィルタによって畳込み処理を行っているため，フィルタサイズ分のデータにフィルタ数を乗じた $(h^2-1)k$ となる．3 層目 (プーリング層) では，2 層目と同様に，プーリングサイズ分のデータにフィルタ数を乗じたものであり，$(s^2-1)k$ となる．4 層目 (全結合層) では，プーリング結果を全結合ノードに集約し，全結合ノードで処理を行うため，プーリング後のサイズに依存する．プーリング後のデータサイズは，畳込み 1 回でサイズが $h-1$ 小さくなり，プーリングで $\frac{1}{s}$ となるため，$\lceil \frac{m-2h+2}{s} \rceil \times \lceil \frac{n-2h+2}{s} \rceil$ となる．したがって，全結合ノードが集約するデータ量は，チャネル数を考慮すると，$(\lceil \frac{m-2h+2}{s} \rceil \times \lceil \frac{n-2h+2}{s} \rceil)k$ となる．順伝搬と逆伝搬で集約データ量は同じであるため，提案手法における集約データ量は，各層でノードが集約するデータ量の和の 2 倍となる．1 層目から 3 層目は入力データサイズに依らず，フィルタ数およびフィルタサイズ等のニューラルネットワークのハイパーパラメータに依存し，このときの集約データ量の和を L とする $(L = (h^2-1) + (h^2-1)k + (s^2-1)k)$．また，プーリング後のサイズを m'，n' とすると，提案手法における，ノードの集約データ量は $2(L + m'n'k)$ となる．

つまり，$mn > 2(L + m'n'k)$ を満たすとき，ノードの集約データ量は，従来の集約型と比較して削減できる．6 章の実験において行ったパラメータ m=17，n=25，k=3，h=3，s=3 の場合におけるフィルタ数 k，フィルタサイズ h，プーリングサイズ s，入力データ $m \times n$ を変更したときのグラフをそれぞれ図 6(a)，図 6(b)，図 6(c)，図 6(d) とに示す．フィルタ数とフィルタサイズが大きい場合，集約データ量も大きく，プーリングサイズが大きければ集約データ量は小さくなる．ネットワークモデルにも依存するが，提案手法におけるネットワークモデルでは，入力データ $m \times n$ が大きくなれば，従来の集約型と提案手法による分散型のノードの集約データ量の差が大きくなる．

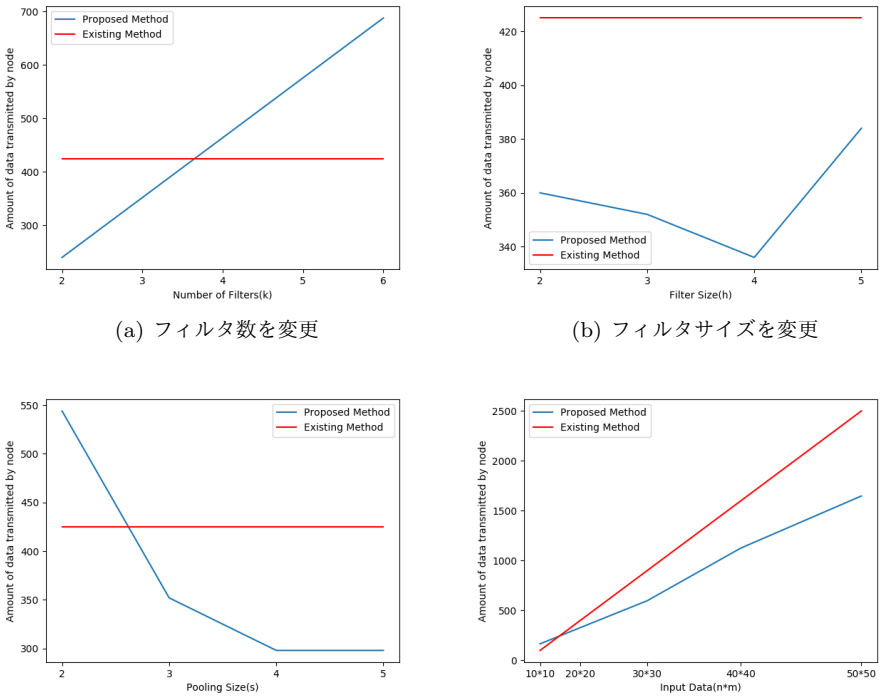
ノードの集約データ量は，ニューラルネットワークおよびそのネットワークにおけるハイパーパラメータに依存するため，適切なネットワークを選択するとともに，適切なハイパーパラメータを選択することが重要となる．

6.　ハイパーパラメータの選択による影響の評価

5 章の議論において，適切なニューラルネットワークおよびハイパーパラメータを選択するという制約の下であれば，従来のゲートウェイに全てのデータを集約させる手法と比較して，提案手法ではセンサーネットワーク内で送受信されるトラフィックの削減が可能であることを確認した．しかし，一般的な深層学習の実行においては，学習のシナリオごとに異なる入力データの特徴などから自由にニューラルネットワークおよびハイパーパラメータを選択することが求められる．そこで，ニューラルネットワークおよびハイパーパラメータの選択を上記の制約条件を満たすように行うと，制約がない場合と比較して深層学習の学習精度が悪化することが考えられる．そこで，本研究では実際の環境において構築したセンサーネットワークが収集した温度データを使用して深層学習を行い，ハイパーパラメータの選択が学習精度に与える影響を評価する．

6.1　学習に使用したデータセット

深層学習の実行にあたり，大阪府内のラウンジスペースに設置した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．このデータセットは，ラウンジの空間を 17×25 のメッシュ状の小領域に分割し，各

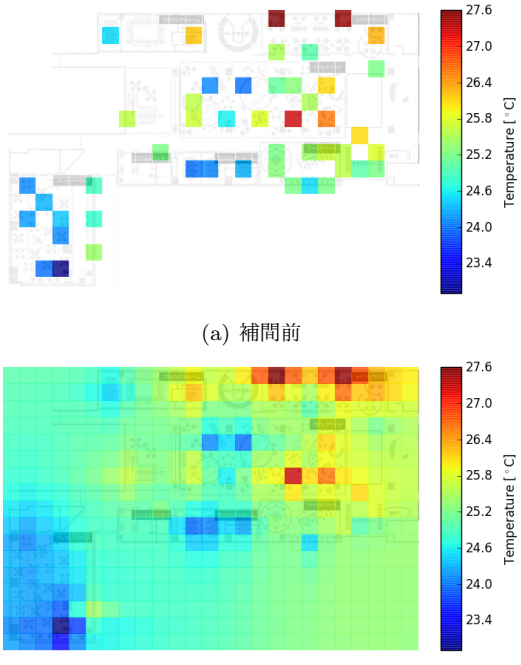


(a) フィルタ数を変更　(b) フィルタサイズを変更

(c) プーリングサイズを変更　(d) 入力データサイズを変更

図 6　ハイパーパラメータの選択に伴うノードの送信データ量の変化

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されている．本研究では 2016 年 8 月 26 日から 10 月 27 日までの間に 30 分間隔で記録された 2961 個の室温マップを深層学習に使用する．なお，ラウンジスペースには温度センサーをおよそ 50 個配置したが，設置場所に制約があったため，センサーをメッシュ状に配置することができなかった．そこで，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することによって入力となる温度の分布データを作成した．補間の方法としては，およそ 50 個程度の温度センサーから取得した実温度データに重みをつける．重みの付け方としては，補間したい位置から実温度データまでのマンハッタン距離をその実温度データの重みとする．そして，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う．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 7(a) と図 7(b) に示す．



(a) 補間前

(b) 補間後

図 7　使用した温度データの一例

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，その空間で異常が起こっているかどうかの正解ラベルが存在しない．そこで，本評価では異常状態を機械的に生成し，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 2961 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，5×5 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，正解となるラベルを付与した．この設定のもとでは，2961 個のデータセットに対

し，およそ 23.84%のデータが異常として判定される．

6.2 ハイパーパラメータを変更した際の学習精度

本節では，制約条件を満たすパラメータ群における学習精度を評価するため，単一の計算機上でニューラルネットワークを構成し，異なるハイパーパラメータの組み合わせにおける学習結果・テスト結果を比較する．

本研究では，用意した 2961 個のデータセットを 2400 個の訓練データと 561 個のテストデータに分割し，バッチサイズ 24，中間層のユニット 200，学習回数 20，パディングサイズ 0 という設定で深層学習を実行した．このような条件の下で，フィルタ数 k，フィルタサイズ h，プーリングサイズ s を変更した際の学習精度をそれぞれ表 1，表 2 および表 3 に示す．なお，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て表に記載している．

表 1，表 2，表 3 に示す結果より，ハイパーパラメータ k，h，s を変更したときの最良のテスト精度は，それぞれ $k = 4, 5$ のとき 91.711%，$h = 5$ のとき 91.214%，$s = 4$ のとき 91.408%であった．また，5 章で述べた，集約データ量を削減するための制約条件を満たしたパラメータは，図 6 より，$k < 3$，$h < 5$，$s > 3$ のときである．この制約条件のもとでの最良のテスト精度は，それぞれ $k = 3$ のとき 90.938%，$h = 5$ のとき 91.214%，$s = 4$ のとき 91.408%であった．この精度は，制約を伴わないハイパーパラメータでの精度と大差がなかった．したがって，ハイパーパラメータの選択に制約条件が存在したとしても，深層学習の性能に与える影響は小さい事が確認できたため，提案手法はセンサー間の通信のトラフィックを抑制しつつデータを処理して知識を得る方法として有効であることが確認された．

表 1 フィルタ数 k を変更したときの深層学習の精度

ハイパーパラメータ (k, h, s)	訓練精度 (%)	テスト精度 (%)
(**2**,3,3)	95.187	90.084
(**3**,3,3)	95.617	90.938
(**4**,3,3)	95.716	**91.711**
(**5**,3,3)	95.820	**91.711**
(**6**,3,3)	95.689	90.572

表 2 フィルタサイズ h を変更したときの深層学習の精度

ハイパーパラメータ (k, h, s)	訓練精度 (%)	テスト精度 (%)
(3,**2**,3)	95.524	90.678
(3,**3**,3)	95.617	90.938
(3,**4**,3)	95.827	90.285
(3,**5**,3)	96.029	**91.214**

表 3 プーリングサイズ s を変更したときの深層学習の精度

ハイパーパラメータ (k, h, s)	訓練精度 (%)	テスト精度 (%)
(3,3,**2**)	95.273	90.019
(3,3,**3**)	95.617	90.938
(3,3,**4**)	96.058	**91.408**
(3,3,**5**)	95.583	89.817

7. おわりに

本研究では，ローカルな無線センサーネットワーク内で深層学習によるデータ解析処理を自律分散的に実行可能なアーキテクチャと，その実装のための深層学習の分散実行アルゴリズムを提案した．今後の課題としては，深層学習を仮想的な環境で集約型と分散型で実行して結果を比較し，重み共有を行わない畳み込み処理における学習精度への影響を検証する．その後，実環境で起こりうるパケットロスおよびセンサーノードの故障などを想定した場合における学習精度への影響を検証する予定である．仮想環境内での深層学習において，分散実行によるノードの集約データ量の削減および学習精度の維持が認められれば，現在ラウンジに展開している IoT センサーによるデータ収集環境に深層学習を分散して実装し，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識別する実証実験を行っていきたい．

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] Lawrence, S., Giles, C. L., Tsoi, A. C. and Back, A. D.: Face recognition: A convolutional neural-network approach, *IEEE transactions on neural networks*, Vol. 8, No. 1, pp. 98–113 (1997).
[2] Wang, P., Li, W., Gao, Z., Zhang, J., Tang, C. and Ogunbona, P. O.: Action recognition from depth maps using deep convolutional neural networks, *IEEE Transactions on Human-Machine Systems*, Vol. 46, No. 4, pp. 498–509 (2016).
[3] Buyya, R., Calheiros, R. N. and Dastjerdi, A. V.: *Big Data: Principles and Paradigms*, Morgan Kaufmann (2016).
[4] Ooi, B. C., Tan, K.-L., Wang, S., Wang, W., Cai, Q., Chen, G., Gao, J., Luo, Z., Tung, A. K., Wang, Y. et al.: SINGA: A distributed deep learning platform, *Proceedings of the 23rd ACM international conference on Multimedia*, ACM, pp. 685–688 (2015).
[5] 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).
[6] Schmidhuber, J.: Deep learning in neural networks: An overview, *Neural networks*, Vol. 61, pp. 85–117 (2015).
[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).