
 解 説

畳み込みニューラルネットワークとその気象観測への応用例

水野 吉規*

要 旨

気象学における機械学習の応用について概観したのち、機械学習のモデルの一つであるニューラルネットワーク、特に畳み込みニューラルネットワークの解説を行う。また、その気象観測に対する応用例として、著者が参加している本庁観測部気象技術開発室（現 大気海洋部業務課気象技術開発室）が進めるカメラ画像からの積雪深推定技術の開発について紹介する。

1. はじめに

科学技術のさまざまな分野において、データに基づいて知識や法則を獲得するデータ駆動 (Data-driven) 型アプローチが注目されている。そしてこれを半ば自動的に遂行する手法として機械学習が分野を越えて大きな関心を集めている。機械学習では、非常に汎用的なモデルをデータにフィットさせることで(これを学習という)、目的とする法則の複製を試みる。その形態は非常に柔軟であり、例えば、入力 $\mathbf{u} \in \mathbb{R}^n$ に対して、 k クラス分類 (Classification) $\mathbb{R}^n \rightarrow \mathbb{R}^k$, 回帰 (Regression) $\mathbb{R}^n \rightarrow \mathbb{R}$ などの機能を実現することができる。これらの他にも、画像や音声中に現れる言葉をテキスト化して出力する転写 (Transcription), 翻訳 (Translation) などへの応用もよくみられる[1]。

気象の予測はモデルの精緻化、計算機能力の向上、そしてデータ同化技術とそれに要する観測データの高水準化を通じて、これまでに大きく発展してきたが[2]、長期の予測や、集中豪雨や竜巻の

ような局所的な極端現象の予測などに課題が残る。これまでは、陽に表わされたモデルに基づく理論駆動 (Theory-driven) 型アプローチが予測精度の向上を主に牽引してきたが、その一方で、各種の観測データや再解析などによるモデルシミュレーションなどから生成される膨大なデータが利用可能となっており、更なる予測精度の向上のためには、データ駆動型アプローチによる知識獲得とモデル確立の必要性が指摘されている[3]。

従来採られてきた理論駆動型アプローチではモデルを陽に表現するためその物理的解釈は容易であるが、データ駆動型アプローチでは得られたモデルの解釈は容易ではない。機械学習によって得られたモデルは、あくまでも学習アルゴリズムの結果として生成されたもので、その物理的意味は必ずしも明らかではなく、そもそも入出力を関係付ける法則の存在を証明するものではない。機械学習が現象の理解にとって有用なものとな

* 気象研究所気象予報研究部
(令和2年12月9日発行)

るためには、学習の結果を人間が解釈できることが重要であるが[4]、その方法論はまだ確立された状況にはなく、今後の更なる研究が待たれる。その一方で、機械学習にはモデルの形態が柔軟で汎用性に富むことや予期せぬ発見があり得ることなどの利点があるため、その物理が未解明な現象に対しても、モデル構築や将来予測のための強力なツールとして期待される。

機械学習が対象とする問題は、古典的にはパターン認識のように空間情報を扱うものと音声認識のように時系列情報を扱うものに分けられるが、これらを組み合わせた手法による動画の解析 (Motion prediction) なども行われる。対象とする問題が形式的に類似することから、場の時間発展を扱う気象の分野においても古くから機械学習を用いた取り組みがみられる。例えば、衛星画像における雲の認識と分類[5]、回帰による土壌タイプの分布の推定[6]、分類と回帰の組み合わせによる二酸化炭素の地表面フラックスの時空間構造の解明[7]、などへの応用が試みられている。

気象の予測に用いられる各種のモデルには、基礎方程式の非線形性あるいは粗視化に起因する多くのパラメタリゼーションが含まれる。それらは物理現象として未解明なものもあり、したがってそれらサブモデルの精緻化が課題として残っている。機械学習の応用として、これらを従来の理論駆動型アプローチではなく機械学習によるモデルに置き換えることが考えられる。この視点では、例えば、降雨対流[8, 9]のパラメタリゼーションに機械学習を応用したケースがある。他にも、接地層における乱流輸送、各種の地表面フラックスを決定する陸面過程のモデリングなど機械学習を活用できる余地が多くあると思われる。地球科学分野における機械学習の活用法については、いくつかの形態が提案されている[3]。

機械学習のモデルの一つにニューラルネットワーク (Neural network) がある。ニューラルネットワークには多くパラメータが存在し、これらは与えられたデータに基づいて、ネットワークが目的とする機能あるいは法則を獲得するように決定される。この手順は、いわゆる非線形最適化問

題を解くことに対応しており、現実的な時間内で解ける問題の規模は使用可能な計算機の能力によって制限を受ける。しかしながら、絶え間ない計算機の能力の向上とともにその制限は徐々に緩和されており、今日では多数の要素からなるネットワークを扱うことが可能になっている。さらに、画像解析に対する畳み込みニューラルネットワーク (Convolutional Neural Network, CNN) の有用性が明らかになったことで[10]、様々な分野においてその応用が活発となった。多数の層からなるネットワークは深層ニューラルネットワーク (Deep neural network), あるいはその学習によるモデルの獲得は深層学習 (Deep learning) と呼ばれる。今日の深層学習の隆盛には、計算機能力の向上、大量のデータへのアクセスを可能とする電子データの記録保存技術及び転送技術の向上の寄与も大きい。

気象学における深層 CNN の近年の応用例としては、極端現象の抽出[11, 12], local climate zone の分類・抽出[13], などがある。国内では、降雪をもたらす雲の抽出[14], 停滞前線の抽出[15], といった取り組みがある。また、雲量の推定[16], 河川の水位の推定[17]といった局所的な気象情報をカメラ画像から推定する試みもある。さらに将来予測に関して、気候パターンの予測[18]やナウキャスト[19]などへの応用例がある。また、上で述べたように、気象モデルの一部をニューラルネットワークに置き換えるといった応用方法も考えられる。

次節以降では、ニューラルネットワーク、特に CNN について基本的な解説を行うとともに、その応用例として積雪深スケールの画像による積雪深の推定の問題を考える。

2. ニューラルネットワーク

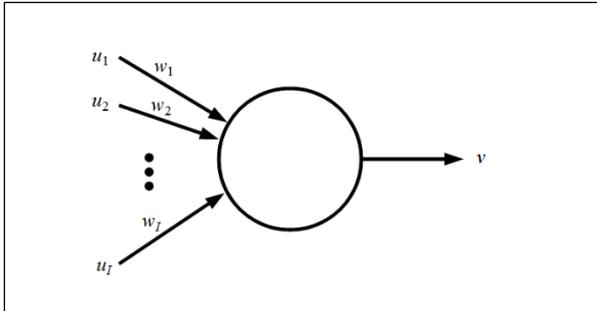
ニューラルネットワークは有向線分によって結合された多くのユニット (ニューロン) から構成される。各ユニットは実ベクトルの入力 $\mathbf{u} = (u_1, u_2, \dots, u_l)$ に対し、次のように 1 つの実数値の出力 v を与える (第 1 図参照)。

$$v = f(w_1 u_1 + w_2 u_2 + \dots + w_l u_l)$$

ここで、 $\mathbf{w} = (w_1, w_2, \dots, w_l)$ は入力に対する実数の重み係数である。 f は活性化関数と呼ばれ、入力の線形結合に対するユニットの応答を表す。 古くはシグモイド関数 $f(x) = 1/[1 + \exp(-x)]$ が用いられたが、最近では Rectified Linear Unit (ReLU) 関数 $f(x) = \max(0, x)$ を用いるのが主流である [20]。 これら関数の定性的な特徴として、引数が 0 以下ではほぼ 0、引数が 0 を越えると正の数を入力する。 ユニットの生物の神経細胞の簡単な数理モデルとして提案されたもので、ユニットが正数を入力している状態は発火状態 (activated) などと表現される。 一つのユニットのはたらきは、入力信号空間 (u_1, u_2, \dots, u_l) を超平面 $w_1 u_1 + \dots + w_l u_l = 0$ によって発火状態と非発火状態の領域に分類することである。 通常は、各ユニットに $u_{l+1} = 1$ のダミー入力を加え、出力を

$$v = f(w_1 u_1 + w_2 u_2 + \dots + w_l u_l + w_{l+1}) \quad (1)$$

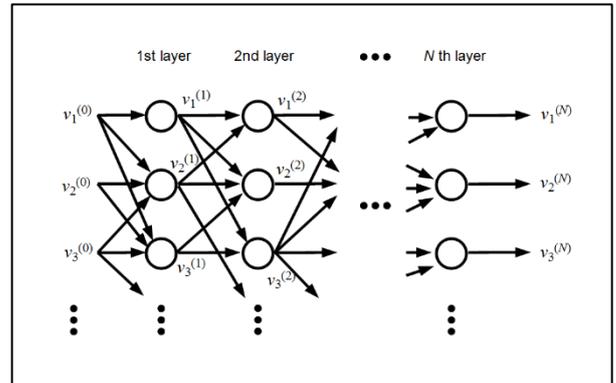
として、発火状態となるしきい値を 0 から変化できるようにする。



第 1 図 ニューラルネットワークを構成するユニット (ニューロン) の概念図。 図中の $\mathbf{u} = (u_1, u_2, \dots, u_l)$ はユニットへの入力信号、 v は出力信号、 w_i は i 番目の入力信号に対する重み係数を表す。

ニューラルネットワークのトポロジーは無数に存在するが、閉路がなく信号が一方方向に伝播していくネットワークを順伝播型ニューラルネットワーク (Feedforward Neural Network, FNN) という。 FNN では、ユニットをそれぞれの入力からの距離 (有向線分の最小数) ごとにグループ分けすることができ、距離 n のユニットのグループを第 n 層と呼ぶ (第 2 図参照)。 最初の層は入力層、最後の層は出力層、中間の層は隠れ層 (Hidden

layers) と呼ばれる。 通常は、FNN は特定の機能を持った層 (モジュール) を直列に組み合わせることで構成する。以降では基本的に FNN を考える。



第 2 図 N 層からなる FNN. 第 n 層のユニットへの入力を $\mathbf{v}^{(n-1)}$ 、出力を $\mathbf{v}^{(n)}$ と表す。

本節では、ニューラルネットワークについて基礎的な解説を行う。本節の構成は次のとおりである。第 2.1 節では分類問題に適用する際の出力層について、第 2.2 節はニューラルネットワークの学習、特に教師あり学習について説明する。第 2.3 節では、FNN の他によく用いられる回帰型ネットワークについて簡単に触れる。最後に第 2.4 節では畳み込みニューラルネットワークについて説明する。

2.1 分類問題における出力層

ニューラルネットワークを分類問題に適用する場合、ネットワークは各クラスのスコアを出力する。スコアは正の実数で、総和が 1 になるように規格化され、どのクラスが尤もらしいかを確率的に表現する。ここでは N クラス分類問題を考えることにすると、出力層のユニット数を N として、それぞれに各クラスのスコア y_1, \dots, y_N を出力させる。出力層の n 番目のユニットに対する入力の線形結合を a_n と表すことにすると、出力 y_n は次のように与えられる。

$$y_n = e^{a_n} / \sum_{n=1}^N e^{a_n} \quad (2)$$

(2) 式右辺の多変数ベクトル値関数はソフトマックス (Softmax) 関数と呼ばれる。特に、 $N = 2$ の場合は、

$$y_1 = \frac{1}{[1 + e^{-(a_1 - a_2)}]}, \quad y_2 = \frac{1}{[1 + e^{-(a_2 - a_1)}]} = 1 - y_1$$

となり，スコアがシグモイド関数で与えられることがわかる．したがって，ソフトマックス関数はシグモイド関数による 2 クラスの分類を 3 以上のクラスの分類に拡張したものと考えることができる．

2.2 教師あり学習

ニューラルネットワークには各ユニットの入力に対する重み係数として多数のパラメータが存在する．これらを決定するための手法として，ここでは教師あり学習 (Supervised learning) について述べる．これは，ネットワークに入力とそれに対応する出力 (教師信号) を与えることで，入力と出力の関係を学習させる方法である．具体的には，教師信号と実際の出力との差を適切に評価した評価関数を統計的に最小化するように各層のパラメータを決定する．一般に評価関数は各パラメータに関して非線形なので，この問題は非線形最適化問題に帰着し，勾配法 (Gradient descent) によって解かれる．勾配法では，評価関数のパラメータに関する偏微分から得られる勾配ベクトルに沿ってパラメータを更新し，評価関数が最小 (極小) 値に達するまでこれを繰り返す．これをシンボリックに表現すると，ある層の重み w は，

$$w \leftarrow w - \alpha \frac{\partial E}{\partial w}$$

によって更新する．ここで， E は評価関数， α は通常は 1 より小さい正の値をとる．少し具体的に，第 2 図の FNN に対する勾配法を考える．ここで第 n 層における重みを $w^{(n)}$ と表わすことにする．出力層における重み係数 $w^{(N)}$ は，

$$w^{(N)} \leftarrow w^{(N)} - \alpha \frac{\partial E}{\partial v^{(N)}} \frac{\partial v^{(N)}}{\partial w^{(N)}}$$

によって更新する．一つ前の層の重み係数 $w^{(N-1)}$ は，

$$w^{(N-1)} \leftarrow w^{(N-1)} - \alpha \frac{\partial E}{\partial v^{(N)}} \frac{\partial v^{(N)}}{\partial v^{(N-1)}} \frac{\partial v^{(N-1)}}{\partial w^{(N-1)}}$$

一般に，出力層から n だけ前の層における重み係数 $w^{(N-n)}$ は，

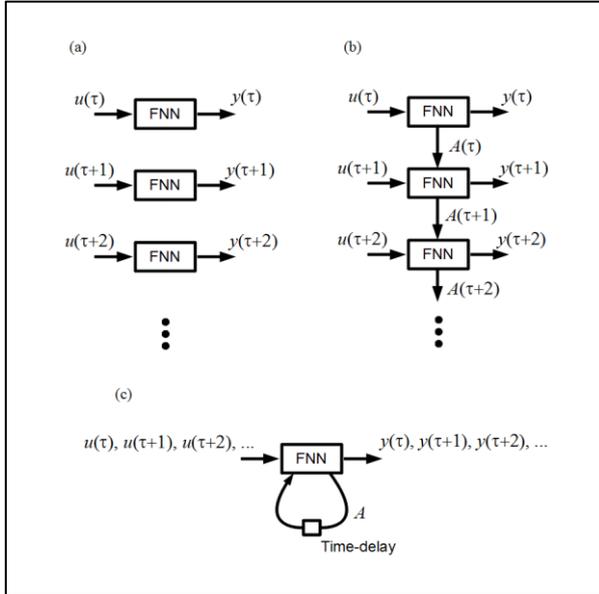
$$w^{(N-n)} \leftarrow w^{(N-n)} - \alpha \frac{\partial E}{\partial v^{(N)}} \frac{\partial v^{(N)}}{\partial v^{(N-1)}} \cdots \frac{\partial v^{(N-n+1)}}{\partial v^{(N-n)}} \frac{\partial v^{(N-n)}}{\partial w^{(N-n)}} \quad (3)$$

によって更新する．(3)式からわかるように，第 $(N-n)$ 層の重み係数を更新する際には， $v^{(N-n-1)}$ 及び第 $(N-n)$ 層以降の情報のみが必要なので，出力側の層の重み係数から順番に更新することになる．このことから，ニューラルネットワークに適用される勾配法は誤差逆伝播法 (Back-propagation algorithm) と呼ばれる．また，勾配ベクトルは全学習データの平均ではなく，一部のデータ (バッチデータ) の平均から求められるものを用いる確率的勾配法 (Stochastic Gradient Descent, SGD) が一般的である．この方法では全学習データの平均として求められる勾配ベクトルに対してランダムネスを付加したベクトルにより更新することになる．この方法により，速やかに適切な重みが得られることが知られている[22]．

2.3 回帰型ニューラルネットワーク (RNN)

気象学では本質的に多次元データの時系列を扱う．本稿では詳しく触れないが，時間発展の予測には，回帰型ニューラルネットワーク (Recurrent Neural Network, RNN) が有用であることが知られている．あるシステムの入力の時系列 $u(\tau), u(\tau+1), u(\tau+2), \dots$ と出力の時系列 $y(\tau), y(\tau+1), y(\tau+2), \dots$ の関係をニューラルネットワークにより予測するためには，第 3 図 a のように各時刻において独立に入出力の関係を予測するのではなく，第 3 図 b のようにシステムの履歴情報を含めて予測することが必要と考えられる．ここで，図中の $A(\tau)$ は隠れ層の一部のユニットの時刻 τ における出力を表し，これが次の時刻 $\tau+1$ における予測の際に一部のユニットへ入力される．このようにシステムの過去の状態を現在の出力の予測に反映させることができるのが RNN である (第 3 図 c)．この RNN では，隠れ層の一部のユニットの出力が時間遅れを経て，一部のユニットへの入力となり閉路を形成している．これにもさまざまなバリエーションがこれまでに提案されているが，中でも，Long Short-Term

Memory networks (LSTMs) [21]は、場の将来予測などに広く用いられている。気象の分野では、例えば、CNN と LSTMs を組み合わせてナウキャストに応用した試みがある[14].



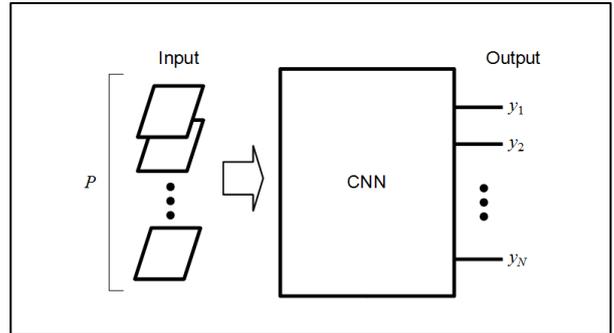
第 3 図 時系列の予測. 時刻 τ における入力を $u(\tau)$, 出力を $y(\tau)$ とする. (a) 時刻ごとに独立した FNN による予測. (b) 前の時刻の状態 A を入力に加えた FNN による予測. (c) (b) を実現する RNN.

2.4 畳み込みニューラルネットワーク (CNN)

CNN は入力データに対して畳み込みによるある種のフィルタリングを行う層 (畳み込み層) を含むニューラルネットワークである. トポロジーは FNN あるいは RNN のいずれでもよい. CNN は特に画像のような場の解析に対して極めて有用であることが知られており, さまざまな問題に活用されている.

CNN を適用する典型的な問題として, D 次元の場のデータから特定のパターンを抽出し, N 個のクラスへ分類する問題を考える (第 4 図参照). システムの入力としてチャンネル数 P の D 次元データを与える. 例えば, 入力データがカラー画像であれば次元数は $D = 2$, 1 ピクセルあたりに必要な情報量は red, green, blue の 3 つなのでチャンネル数は $P = 3$ となる. 第 2.1 節で述べたように, 出力は各クラスのスコアを与える N 要素の 1 次元ベクト

ル $\mathbf{y} = (y_1, y_2, \dots, y_N)$ である. 以下では, CNN を構成する代表的なモジュールについて述べる.



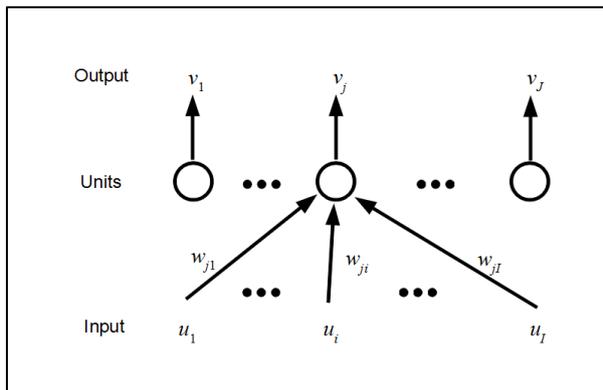
第 4 図 CNN による N クラス分類. チャンネル数 P の場のデータを入力とし, N 個のクラスに対するスコア $\mathbf{y} = (y_1, y_2, \dots, y_N)$ を出力する.

2.4.1 全結合層

まず全結合層について述べる. 層への入力を $\mathbf{u} = (u_1, \dots, u_I)$, 層のユニットの数を J , それぞれからの出力をまとめて層の出力を $\mathbf{v} = (v_1, \dots, v_J)$ と表わす (第 5 図参照). このとき全結合層では, 出力を

$$v_j = f\left(\sum_{i=1}^I w_{ji}u_i\right) \quad (4)$$

で与える. ここで, w_{ji} は重み係数, f は活性化関数である. 全結合層では, 全てのユニットに対し全入力が渡される. 全結合層のパラメータとしてユニット数 J と $J \times I$ 個の重み係数 w_{ji} があるが, 通常はユニット数を固定し, 重み係数を第 2.2 節で述べた勾配法により決定する. 全結合層の機能は, 各ユニットが与える J 通りの超平面による発火/非発火の分類を並列して出力することである. 全結合層は CNN に限らず汎用的に用いられるモジュールである.



第 5 図 全結合層. $\mathbf{u} = (u_1, \dots, u_l)$ を入力, $\mathbf{v} = (v_1, \dots, v_j)$ を出力, w_{ji} を j 番目のユニットへの i 番目の入力に対する重み係数とする.

2.4.2 畳み込み層

畳み込み層は Q 個のフィルタを持ち, チャンネル数 P の D 次元データの入力に対して, チャンネル数 Q の D 次元データを出力する. ここでは 2 次元 (画像) データを考え, 入力を $u_{m,n,p}$, 出力を $v_{m,n,q}$ と表すことにする. ここで, $m = 1, \dots, M$, $n = 1, \dots, N$, $p = 1, \dots, P$, $q = 1, \dots, Q$ である. また, q 番目のフィルタのカーネルを $w_{s,t,q}$ と表すことにする. ここで, $s = -S, \dots, S$, $t = -T, \dots, T$ である. したがって, 畳み込み層には $M \times N \times Q$ のユニットとそれらへの入力に対する総数 $(2S + 1) \times (2T + 1) \times Q$ の重み係数が存在する. 2 次元データに対する畳み込み層の概略図を第 6 図に示す. このとき, 出力を畳み込み和

$$v_{m,n,q} = f \left(\sum_{p=1}^P \sum_{s=-S, t=-T}^{S, T} w_{s,t,q} u_{m+s, n+t, p} \right) \quad (5)$$

で与える. f は活性化関数であり, 先に述べた ReLU 関数を用いられるのが主流である. 各 q に対する出力 $v_{m,n,q}$ で表される 2 次元データは feature map と呼ばれ, これをさらに次の畳み込み層への入力とすることもできる.

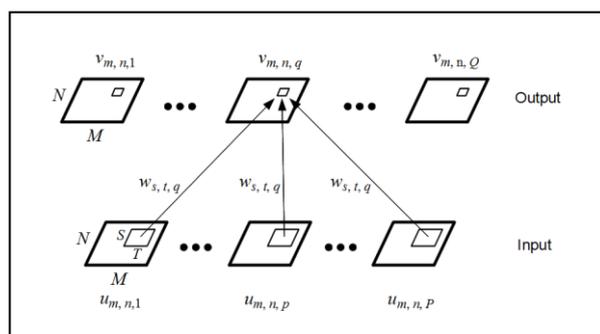
フィルタのカーネルを形成する重み係数 $w_{s,t,q}$ が畳み込み層におけるパラメータであり, 学習によって決定される. ただし, ネットワークのトポロジーを固定するため, カーネルのサイズと枚数は予め決めておく. 学習により適切に設定されたフィルタは入力の map からターゲットに関する

何らかの特徴だけを抽出するはたらきをするものと期待される. 例えば, 次のような大きさ 3×3 のカーネル

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

は画像に対して横方向の差分をとることによって, 色合いや明るさの急激な変化を検出する. このカーネルを用いたフィルタリング操作により作成される feature map は画像に現れるパターン of 縦方向の輪郭を抽出した画像となる. 輪郭を余すことなく得るためだけでも, 様々な方向の差分をとる複数枚のカーネルが必要となる. Feature map によるターゲットの特徴抽出の具体的な例は [23]に見られる. 畳み込み層を何層にも重ねることで, 段階的に元のデータを抽象化していき最終的に分類に重要な特徴量の抽出を行うのが CNN の基本的な機能である.

なおここでは簡単のため, (5) 式の畳み込み和は全てのピクセルに対して計算し, 入力と出力の map の解像度は同じとしている. 実際には, 畳み込み和を計算するピクセルを等間隔に間引いたり, map の端を省いたりして, 出力される map の解像度を入力よりも小さくする場合もある.



第 6 図 2 次元データに対する畳み込み層. $u_{m,n,p}$ は入力, $v_{m,n,q}$ は出力, $w_{s,t,q}$ はフィルタのカーネルである. ここで, $m = 1, \dots, M$, $n = 1, \dots, N$, $s = -S, \dots, S$, $t = -T, \dots, T$, $p = 1, \dots, P$, $q = 1, \dots, Q$ である. マップのサイズは (M, N) , カーネルのサイズは $(2S + 1, 2T + 1)$ である.

2.4.3 プーリング

さらに、畳み込み層から出力された feature map に対して、プーリングと呼ばれる処理がなされることが多い。これはターゲットのパターンについて、ある種の変換に対して不変な特徴を抜き出すことを目的とする。例えば、入力された feature map を小領域に分割し、それぞれの小領域における代表値を出力する処理がよく用いられる。代表値として、その小領域の最大値を出力する場合は最大プーリング (Max pooling)、平均値を出力する場合は平均プーリング (Average pooling) と呼ばれる[1]。これらの処理はいずれも map の粗視化を行うことに対応しており、分割された小領域のサイズがそれより小さい範囲の平行移動や変形に対して不変な map を生成する。また、粗視化によるノイズの除去と解像度の低減は、過学習の防止や学習に伴う計算量の軽減にも寄与する。小領域のサイズは目的に応じて適切に設定する必要がある。プーリングを一部のチャンネルにのみ適用するなど、その運用にはさまざまなバリエーションがあり得る。

古典的な CNN は複数の畳み込み層とそれに続く少数の全結合層から構成される。例えば、CNN が再注目されるきっかけとなったネットワーク[10]は 5 層の畳み込み層とそれに続く 2 層の全結合層から構成されている。現在はさらに多層化・複雑化したネットワークが用いられている。

深層 CNN の詳細については、書籍[1]、あるいはレビュー論文[24]を参照されたい。また、黎明期から最近までのニューラルネットワーク技術の進展について詳しく述べられた文献[25]もある。興味のある読者はこちらも参照されたい。

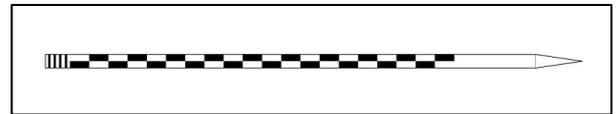
3. CNN の応用例ーカメラ画像による積雪深の推定のための予備実験ー

ここでは、監視カメラによって撮影された画像から積雪深を推定する問題を考える[26]。これはアメダスに設置された環境監視カメラの画像から積雪量を推定することを想定したもので、積雪深計の整備されていない地点における補助的データの提供を目的としている。CNN を用いて、積

雪により一部が埋没したスケールのパターンをカメラ画像から読み取ることで、積雪深を精度よく推定することが最終的な目標である。

実際のカメラ画像に現れるスケールは、降雨、降雪、明るさなどの気象条件、あるいは、スケールに付着するよごれ、雪氷などによって一つとして同じように見えるものはない。さらに、カメラを設置するサイトによってカメラに対するスケールの位置が異なる。これら様々な要因に対してロバストな推定システムを設計する必要がある。ここでは、カメラ画像におけるスケールの幾何変化が積雪深の推定精度に与える影響についての予備調査の結果を紹介する。

本予備実験では、第 7 図に示す積雪深スケールの仕様を想定した。スケールの幅は 38 mm で、地表から 100 mm までの部分には左右非対称の白黒パターンを塗装してあり、この部分で積雪深を測る。一つの縞の間隔は 50 mm である。またスケール最上部には目印として幅 12 mm 程度の 9 本の縞模様の塗装を施してある。



第 7 図 積雪深スケール。右端を地面に挿入し鉛直に設置する。

カメラ画像に現れる積雪深スケールは、カメラとの位置関係やレンズの性質により、以下のようにその形状などが変化する。

位置：カメラに対するスケールの位置はサイトごとに異なる

遠近効果：カメラは地面に対して斜めに見下ろすように設置されるため、カメラ画像におけるスケールではその上端から下がるにつれて幅や縞模様の間隔が狭くなる。

傾き：スケールがカメラの正面になければ、スケールは傾斜して見える。

レンズ歪み：レンズの歪みにより、スケールが変形する。

ニューラルネットワークによる積雪深の推定を行う際に、これらの効果が推定精度に与える影

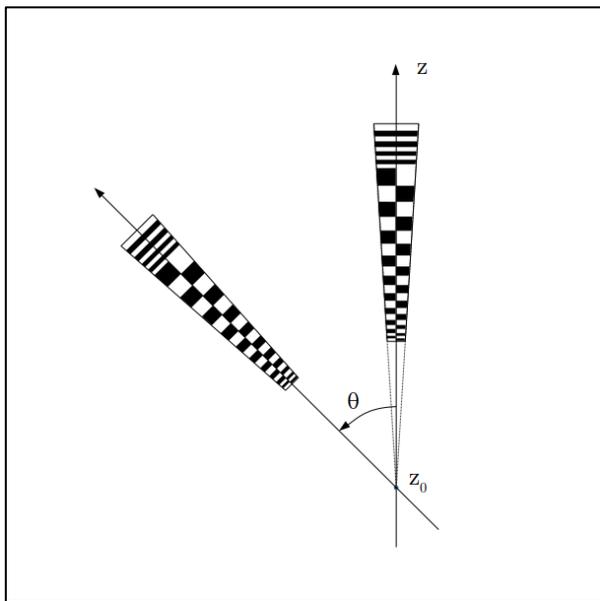
響を、以下に述べる予備実験によって調査した。

3.1 画像モデル

学習及び評価に用いるカメラ画像はコンピュータグラフィックスによって作成した。上で述べたスケールの幾何変化を表現するために使用したモデルは以下のとおりである。

遠近効果によって、積雪深スケールはある点（以下、消失点と呼ぶ）を頂点とする逆二等辺三角形の一部として現れ、積雪深スケールの幅や縞模様の間隔が地面からの高さに対して線形的に変化する（第 8 図）。ここでは、画像下端からの消失点の高さ z_0 をパラメータとする。 z_0 の変化に伴ってスケールの鉛直方向の拡大率が変化する。

スケールの傾きは、消失点まわりに積雪深スケールを回転させることで表現される（第 8 図）。回転角 θ がパラメータとなる。またこの操作によって、消失点の位置 z_0 と回転角 θ に依存して積雪深スケールの位置が変わる。



第 8 図 積雪深スケールの消失点 z_0 と回転角 θ

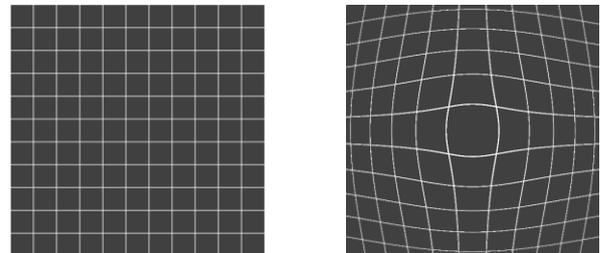
レンズ歪みに対しては、画像中心を原点とする半径方向の歪み (Radial lens distortion) を考える。一般的には、歪みを偶数次の多項式で与えるモデルが用いられるが[27]、ここでは簡単のために、ある基準点（通常は画面の中心）からの距離 r におけるピクセル情報が、写像

$$r \mapsto R (r/R)^\beta \quad (6)$$

によって移動する歪みモデルを用いた。ここで、 R は基準点からの画像内の最大距離、 $\beta \in (0,1)$ は拡大率を表すパラメータである。このモデルによる格子模様の歪みの例を第 9 図に示す。この例では、基準点は画像の中心、 R は画像の対角線の長さの半分、 β は 0.7 とした。 (6) 式で表されるモデルは、基準点付近では放射状に拡大、遠方では縮小されるような歪みを与えることがわかる。このような性質の歪みは barrel distortion と呼ばれ、広角レンズによる撮影画像において典型的に現れる。Radial lens distortion の補正に CNN を応用するケースもあるが[28]、ここでは陽に歪みの補正は行わない。

画像を作成する手順は以下のとおりである。

0. 雪表面より上の部分の長方形のスケールを描画（位置がパラメータ）
1. 消失点の位置 z_0 を与え、スケールを逆二等辺三角形に変形（ z_0 がパラメータ）
2. 消失点周りに θ だけ回転（ θ がパラメータ）
3. (1) 式によりスケールを変形（ R と β がパラメータ）



第 9 図 (6) 式によるひずみの例。左の画像を (6) 式によって変換すると右の画像になる。基準点は画像の中心、 R は画像の対角線の長さの半分、拡大率は $\beta = 0.7$ とした。

3.2 ニューラルネットワークによる推定

前節で述べた幾何変化を異なる組み合わせで与えた 5 つのケース（第 1 表）に対して、同一のシステムにより積雪深の推定を行った際の推定精度の比較を行った。ここでは、積雪深スケールの非対称縞の数である 20 クラス分類問題をニューラルネットワークによって解いた。

第 1 表 実験を行ったケース. 作成手順の列には, サンプル画像を作成する際に行った手順 (第 3.1 節参照) を示す. 手順 1 により遠近効果, 2 により傾き, 3 によりレンズ歪みの効果が加味される.

	作成手順	遠近効果	傾き [deg]	歪み	データ数 (解像度)
Case 1	0, 2	無し	$\theta = 45$	無し	60,000 (160×180)
Case 2	0, 1	有り	$\theta = 0$	無し	60,000 (64×180)
Case 3	0, 1, 2	有り	$\theta = 45$	無し	60,000 (160×180)
Case 4	0, 1, 2, 3	有り	$\theta = 45$	有り	60,000 (160×180)
Case 5	0, 1, 2, 3	有り	$\theta = -45-45$	有り	40,000 (320×225)

i 番目 ($i = 1, 2, \dots, 20$) のクラスを雪表面が $(i-1)/20$ m から $i/20$ m の範囲にあることに対応させ, $1/20$ m の分解能で積雪深を推定することを考える. 教師信号 $\mathbf{t} = (t_1, \dots, t_{20})$ は, 積雪深が $(i-1)/20$ m から $i/20$ m にあると判定された場合は $t_i = 1$, それ以外の要素は 0, となるベクトルで与える.

5 つの各ケースにおいて学習及び評価に用いた画像のサンプルを第 10 図に示す. いずれの画像においても積雪深スケールの最大幅は 8 ピクセル程度となっている. 想定される実際の画像の解像度に比べるとかなり粗いが, 学習に伴う計算量の軽減のためにこのような低解像度での評価を行った. スケールの消失点の高さ z_0 は, スケール下端から下側にスケール全長の 3 分の 1 の範囲で一様乱数によりランダムに与えた. Case 5 ではスケールの傾き角を -45° から 45° までの範囲で一様乱数によりランダムに与えた. また, case 4, 5 では歪みのパラメータ β は 0.7 に固定した. ただし, スケールの位置がサンプルごとに異なるので, 歪みによるスケールの変形もサンプルごとに異なる.

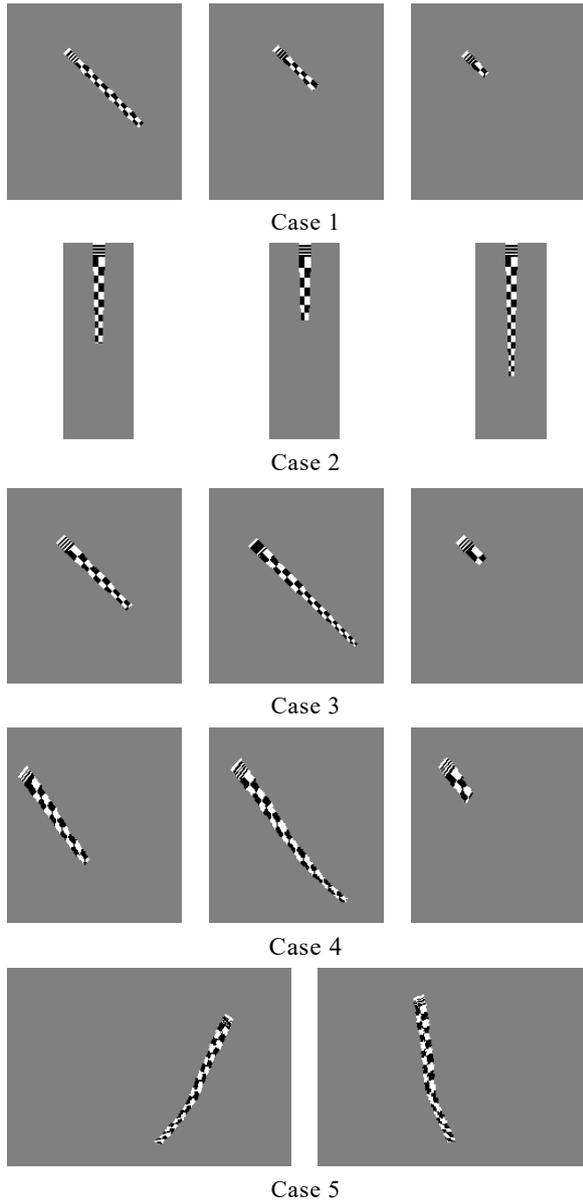
- 畳み込み層 (フィルタ数 4, カーネルサイズ 8×8)
- 畳み込み層 (フィルタ数 4, カーネルサイズ 8×8), dropout (無効化率 0.5)
- 全結合層 (ユニット数 256), dropout (無効化率 0.5)
- 全結合層 (ユニット数 20)

から構成されるネットワークを用いた. 畳み込み層におけるフィルタのカーネルのサイズは画像

に現れるスケールの最大幅程度を目安として 8×8 ピクセルとした. ここで紹介するフィルタ数及び 1 層目の全結合層のユニット数は, おおむね最も良好な結果が得られたものであるが, これらに関する最適化は行っていない. 最後の全結合層が出力層であり, 20 のクラスに対するスコア $\mathbf{y} = (y_1, \dots, y_{20})$ (ここで, $0 < y_i < 1$, $\sum_i y_i = 1$) を出力する. 2 層目の畳み込み層と 1 層目の全結合層では無効化率 0.5 の dropout [29] を適用している. 畳み込み層及び 1 層目の全結合層における活性化関数は ReLU 関数, 出力層では, 第 2.2 節で述べた, Softmax 関数を用いた. 学習には先に述べた SGD を洗練させた Adam 法 [30] を用い, 評価関数は交差エントロピー誤差

$$E(\mathbf{t}, \mathbf{y}) = - \sum_{i=1}^{20} t_i \log y_i \quad (7)$$

を用いた. いずれのケースにおいても, 全データの 8 割を学習データ, 残りの 2 割を評価データとして用いた. 全学習データに対する誤差の平均値が最小 (極小) となるように, 重み係数を決定する. (先に述べた SGD では, 学習データをバッチデータに分割し, 各バッチデータを用いた学習を順次おこなう.) 評価データは学習によって得られた重み係数を擁するシステムが未知のデータに対してどの程度の誤差を与えるかを評価するのに用いられる. 通常はこの学習・評価の手続きを, 評価データにより評価された誤差が収束するまで繰り返し行う. この反復数はエポック数などと呼ばれる. また, 誤差と同時に $\sum_i t_i y_i$ の平均として定義される正答率も評価する.



第10図 画像の例. 上から順番に, case 1 から case 5. レンズ歪みの基準点は, case 4 では画像の右下, case 5 では下端中心.

ここで用いている交差エントロピー誤差 (7) 式は, スケールの読み取り値の誤差を評価するものではないことに注意しなければならない. 例えば, 正解がクラス 1 (積雪深 0 cm-5 cm) の画像に対して, クラス 2 (積雪深 5 cm-10 cm) を推定した場合とクラス 3 (積雪深 10 cm-15 cm) を推定した場合を比較すると, 読み取り値の誤差は前者の方が小さいことになるが, 交差エントロピー誤差 (7) 式ではこれらは同等に扱われる. 読み取りの誤差を評価するためには, クラス間の距離を導入

し評価関数を定義しなおす必要があるが, ここではこれ以上立ち入らない. 以下では, 正解のクラスを推定する確率である正答率を用いてシステムの性能の評価を行うことにする.

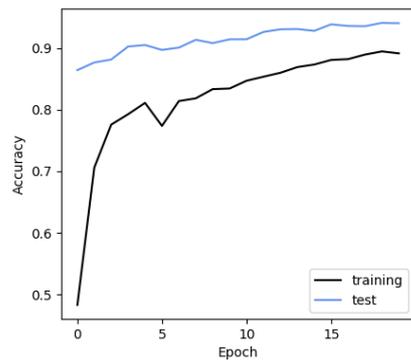
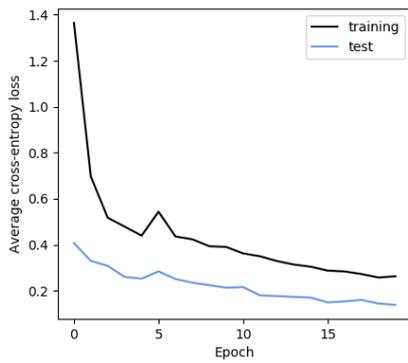
以上のシステムの実装には, Python 3.7.3 とライブラリ TensorFlow 1.14 [31]を用いた. 学習ではバッチサイズを 200 とした.

3.3 結果

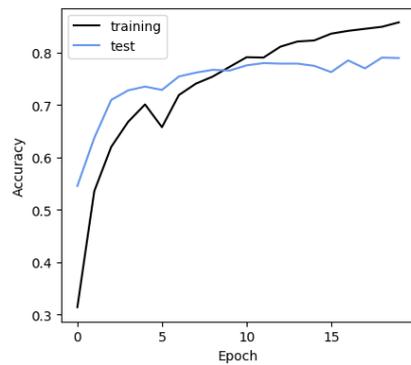
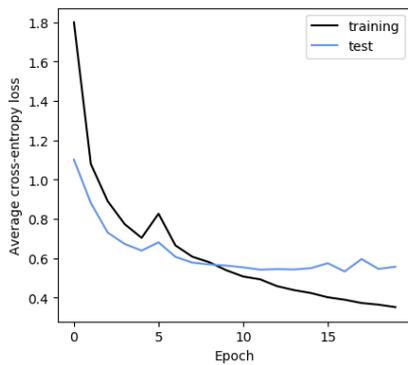
Case 4 と 5 に対する学習経過の例を第 11 図に示す. いずれの場合も, 20 回程度の学習で評価における誤差, 正答率が収束している様子がわかる. 通常は評価データに対する正答率は学習データよりも小さくなるが, case 4 に対しては, 中間の層に dropout を適用することにより, 評価データに対しての方が学習時よりも高い正答率が得られていることがわかる. 実験の結果, case 1 から case 4 の正答率はいずれも 95%程度となり, 積雪深スケールの遠近効果, レンズ歪みの有無, スケールの傾きの有無が推定精度に与える影響は小さいことがわかった. その一方で, 積雪深スケールの傾きがランダムな唯一のケースである case 5 の正答率は 80%程度であった. このことは, 傾きのばらつきは推定を困難にすることを示している.

なお, 計算機環境の制約のため case 5 の学習データ数が他に比べて少ないが, case 1 から case 4 に与えるデータ数を case 5 と同じ 40,000 としても, 正答率は 1%-3%程度低下するのみであった. したがって, 上の結論は学習データ数の違いに起因するものではないことも確認している.

以上の結果から, カメラ画像における積雪深スケールの位置や形状の変化に対して積雪深の推定はロバストに行うことができるが, スケールの傾きの変化に対してはその正答率が低下することがわかった. この対策としては, (1) 回転に対するプリーング処理を行い[32], 回転に対して頑健なシステムを構築する. (2) 前処理により回転などを補正する. (3) それぞれのサイトの画像ではスケールの向きは変わらないので, サイトごとに学習させる. などが考えられる.



Case 4



Case 5

第 11 図 Case 4, 5 における学習の経過の例. 左: 平均交差エントロピー誤差, 右: 正答率. 横軸はエポック数. Training と test はそれぞれ学習データと評価データに対する誤差と正答率の経過を表わす. 誤差の減少に伴い, 正答率が増加していることがわかる.

なお, 環境監視カメラによる積雪推定は運用を目指して現在も開発が進行中である. これまでの成果は[26]を参照されたい. また, 同様のカメラは地表面の様々な情報を捉えることができるため, 例えば, 植生の様子から地表面の粗度や蒸発散量などの推定するなどに活用することも考えられる.

4. 終わりに

本稿では, 近年注目されている畳み込みニューラルネットワークについて簡単に解説し, その積雪深推定への応用について紹介した. ここでは基本的な事柄にしか触れておらず, ニューラルネットワークによる深層学習は現在も発展途上にある. この手法自体は汎用的であるので, 分野にとらわれず様々な問題に適用できることが期待さ

れる. また, AlexNet, GoogleNet などさまざまなモデルが公開されているため, 非専門家でも容易にニューラルネットワークの実装が可能な環境にある. 気象の分野においても, 問題解決のための一つの手法として機械学習の利用は今後も拡大していくと思われる[33, 34].

本稿の執筆にあたって有益なご助言をいただいた気象庁観測部気象技術開発室(現 大気海洋部業務課気象技術開発室)の小林健二氏, 北村智文氏に感謝いたします.

参考文献

- [1] Goodfellow, I. *et al.* (2016) : Deep learning. MIT Press, Cambridge.
- [2] Bauer, P. *et al.* (2015) : The quiet revolution of numerical weather prediction. *Nature*, **525**, 47-55.
- [3] Reichstein, M. *et al.* (2019) : Deep learning and process understanding for data-driven Earth system science. *Nature*, **566**, 195-204.
- [4] Montavon, G. *et al.* (2017) : Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, **73**, 1-15.
- [5] Lee, J. *et al.* (1990) : A neural network approach to cloud classification. *IEEE Transactions on Geoscience and Remote Sensing*, **28**, 846-855.
- [6] Grimm, R. *et al.* (2008) : Soil organic carbon concentrations and stocks on Barro Colorado Island – digital soil mapping using Random Forests analysis. *Geoderma*, **146**, 102-113.
- [7] Papale, D. and R. Valentini (2003) : A new assessment of European forests carbon exchanges by eddy fluxes and artificial neural network spatialization. *Global Change Biology*, **9**, 525-535.
- [8] Schirber, S. *et al.* (2013) : Parameter estimation using data assimilation in an atmospheric general circulation model: from a perfect toward the real world. *Journal of Advances in Modeling Earth Systems*, **5**, 58-70.
- [9] Gentine, P. *et al.* (2018) : Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, **45**, 5742-5751.
- [10] Krizhevsky, A. *et al.* (2012) : Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, **25**, 1097-1105.
- [11] Liu, Y. *et al.* (2016) : Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *Proceedings of International Conference on Advances in Big Data Analytics*, 81-88.
- [12] Racah, E. *et al.* (2017) : ExtremeWeather: a large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. *Advances in Neural Information Processing Systems*, **30**, 3405-3416.
- [13] Yoo, C. *et al.* (2019) : Comparison between convolutional neural networks and random forest for local climate zone classification in mega urban areas using Landsat images. *ISPRS Journal of Photogrammetry and Remote Sensing*, **157**, 155-170.
- [14] 鈴木ほか (2019) : ディープラーニングによる南極昭和基地周辺における降雪をもたらす雲の検出. 雪氷研究大会講演要旨集, 119.
- [15] Matsuoka, D. *et al.* (2019) : Automatic detection of stationary fronts around Japan using a deep convolutional neural network. *SOLA*, **15**, 154-159.
- [16] Onishi, R. and D. Sugiyama (2017) : Deep convolutional neural network for cloud coverage estimation from snapshot camera images. *SOLA*, **13**, 235-239.
- [17] 前原ほか (2019) : ディープラーニングに基づく CCTV カメラ映像からの水位計測方法. 「写真測量とリモートセンシング」, **58**, 28-33.
- [18] Chattopadhyay, A. *et al.* (2020) : Predicting clustered weather patterns: a test case for applications of convolutional neural networks to spatio-temporal climate data. *Scientific Reports*, **10**, 1317.
- [19] Shi, X. *et al.* (2015) : Convolutional LSTM network: a machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, **28**, 802-810.
- [20] Glorot, X. *et al.* (2011) : Deep sparse rectifier networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*. PMLR, **15**, 315-323.

- [21] Hochreiter, S. and J. Schmidhuber (1997) : Long short-term memory. *Neural Computation*, **9**, 1735-1780.
- [22] Bottou, L. and O. Bousquet (2007) : The tradeoffs of large scale learning. In *Proceedings of Advances in Neural Information Processing Systems*, **20**, 161-168.
- [23] LeCun, Y. *et al.* (1990) : Handwritten digit recognition with a back-propagation network. In *Proc. Advances in Neural Information Processing Systems*, 396-404.
- [24] LeCun, Y. *et al.* (2015) : Deep learning. *Nature*, **521**, 436-444.
- [25] Schmidhuber, J. (2015) : Deep learning in neural networks: An overview. *Neural Networks*, **61**, 85-117.
- [26] 北村ほか (2019) : アメダス観測環境監視カメラから積雪深を推定するアルゴリズムの開発. *信学技報*, **119**, 19-24.
- [27] Heikkilä, J. and O. Silvén (1997) : A four-step camera calibration procedure with implicit image correction. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, USA, **1997**, 1106-1112.
- [28] Rong, J. *et al.* (2017) : Radial lens distortion correction using convolutional neural networks trained with synthesized images. *Computer Vision – ACCV2016* (eds. Lai SH *et al.*), **10113**, 35-49, Springer.
- [29] Srivastava, N. *et al.* (2014) : Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, **15**, 1929-1958.
- [30] Kingma, D. P. and J. L. Ba (2014) : Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [31] Abadi, M. *et al.* (2016) : Tensorflow: a system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating System Design and Implementation*, **16**, 265-283.
- [32] 森山ほか (2017) : 2次元画像における面内回転不変な物体認識に向けて. *情報処理学会研究報告*, 2017-CVIM-207.
- [33] 三好建正 (2018) : 人工知能 (AI) は気象学にブレークスルーをもたらすか? *日本気象学会 2018 年度秋季大会講演予稿集*, D351.
- [34] 三好建正 (2019) : 気象学における AI 活用から AI 統合へ. *日本気象学会 2018 年度秋季大会講演予稿集*, B351.