

# 点群深層学習を用いた 法面の異常検知手法の開発

石川 大智<sup>1</sup>・貝戸 清之<sup>2</sup>

<sup>1</sup>学生会員 大阪大学大学院工学研究科 地球総合工学専攻 (〒 565-0871 吹田市山田丘 2-1)  
E-mail: t.ishikawa@civil.eng.osaka-u.ac.jp

<sup>2</sup>正会員 大阪大学准教授 大学院工学研究科 地球総合工学専攻 (〒 565-0871 吹田市山田丘 2-1)  
E-mail: kaito@ga.eng.osaka-u.ac.jp

斜面・法面のような自然公物の場合、劣化過程に介在する様々な不確実性の影響が大きく、専門技術者であってもそれらの異常を早期に検知することが難しい。このような状況下では、空間的に高密度な点群データを用いた異常検知に頼らざるをえない。社会基盤工学分野においては、MMS に代表される点群データ計測のためのレーザー計測技術が発展している。しかし、点群データは地物ごとの変状や時系列変化を容易に把握できないために、未だ実務における意思決定に十分に活用されていない。そこで本研究では、点群データを入力とする点群深層学習に基づいた斜面・法面の局所的な異常検知手法を提案する。さらに、実際の法面を対象として、変形やはらみ出し等の局所的異常を有する法枠の検知を試みることによって提案手法の有効性を実証的に検証する。

**Key Words :** *point cloud, anomaly detection, deep learning, autoencoder, slope*

## 1. はじめに

近年の我が国では、高度経済成長期に建設された社会基盤施設の急速な老朽化が進んでおり、それらの維持管理が喫緊の課題となっている。とりわけライフサイクル費用<sup>1),2)</sup> やリスク評価<sup>3),4)</sup> の観点から予防保全が重視されることが少なくなく、徹底した予防保全の実現には社会基盤施設の早期異常検知が不可欠である。橋梁を例にあげると、経験豊富な専門技術者の暗黙知、あるいは目視点検データを用いた統計的劣化予測手法<sup>5),6)</sup> によって、劣化進展部材の特定や寿命評価が可能であり、早期異常検知を行うことは技術的には不可能ではない。これは橋梁が人工公物であることから、構造上の弱点部位や部材が特定され、またそれらの劣化メカニズムが解明されている場合が多く、有形無形の知見を利用可能であることに起因する。一方、本研究で対象とする斜面・法面の場合、背後にある地山は自然公物であり、その構造特性や劣化過程などに介在する様々な不確実性の影響が大きい。専門技術者であっても、要監視箇所の抽出、劣化予測や異常検知が困難である。また、点検は防災を目的とした道路防災点検と防災カルテ点検、維持管理を目的とした特定土工点検に大別され、これまで前者に軸足が置かれていた。したがって、健全度判定や劣化予測を目的としたデータや知見が蓄積されていないのが現状である。

異常の発生箇所や時期、劣化進展メカニズムなどが不明な場合には、対象となる構造物に対して、時間的

に長期間、空間的に高密度な計測が必要である。特に斜面に関しては長期モニタリング技術に関する研究蓄積がある<sup>7),8)</sup>。しかし、従来はセンサー設置数の制約から、センサー設置箇所を絞り込む必要があり、予め構造的な弱点や異常発生可能箇所を特定しておかなければならず、適用には高度な判断が要求された。一方、空間的に高密度な計測に関しても、近年点群データの計測技術の発展が著しい。しかし、点群データを用いた異常検知に関する研究は、筆者の知る限りほとんど行われていない。点群を入力データとして扱う場合、4. で述べるような点群固有の性質や高次元であることを考慮する必要があり、時系列データや画像データと比べ扱いにくい。しかしながら、深層学習の分野において、点群データを入力として、分類や物体検出を行う点群深層学習と呼ばれる技術が開発されている<sup>9)–15)</sup>。点群深層学習は未だ発展途上の技術であるが、点群における異常検知モデルの作成においても、点群深層学習における種々の要素技術は有効であると考えられる。

以上の問題意識のもと、本研究では斜面・法面の異常検知を目的として、深層学習に基づく大規模点群を処理可能な異常検知の方法論を開発する。具体的には、深層学習モデルの1つである AutoEncoder に対して、点群データを直接入力として取り込めるように拡張し、斜面・法面における局所的異常を検知するモデルを開発する。以下、2. で本研究の基本的な考え方を説明する。3. では、ニューラルネットワークに関する前提知識を説明した後、点群深層学習における点群固有の性

質を考慮した要素技術を説明する。4. で点群深層学習に基づく異常検知モデルを構築する。5. では、実際の法面において計測された点群データを用いて提案モデルの実証分析を行う。

## 2. 本研究の基本的な考え方

### (1) 社会基盤施設の維持管理と点群データ

近年の我が国では、高度経済成長期に建設された社会基盤施設の急速な老朽化が進んでおり、それらの維持管理が喫緊の課題となっている。また、国家予算に占める社会基盤施設関連の事業費は減少傾向にあり、適切かつ効率的な維持管理を行うことが求められている。これに加えて、土木業界の人材不足と技術・技能の伝承に関する問題<sup>16)</sup>が依然とし解消されず、上記の問題に拍車をかけているのが現状である。一方で、情報分野では近年の計算機の処理能力向上や IoT (internet of things) の進展に伴い、ビッグデータと呼ばれる大規模なデータ集合の活用が推進されており、ビッグデータの活用は社会基盤工学分野における維持管理やモニタリングにも広がっている<sup>17)</sup>。

このような背景のもと、ICT (information and communication technology, 情報通信技術) により建設生産システムにおける生産性の向上を図る i-Construction が推進されている。国内外における、社会基盤施設に対する AI 技術の活用の流れはその一環である。AI の中でも近年急速に発展を遂げている深層学習と呼ばれる技術は種々の分野で革新的な成功を収めており、画像認識、音声認識、機械翻訳などといった様々な分野において、従来の数理モデルでは対処が困難であった問題を解決してきている<sup>18)</sup>。社会基盤工学分野においても、コンクリートのひび割れ検出に代表される画像認識<sup>19)</sup>や舗装路面の劣化予測モデル<sup>20)</sup>などの研究蓄積が確認できる。

一方で、i-Construction による設計から維持管理に至るまでの高度化の流れに伴って、レーザー計測技術が発展しており、座標情報や色情報を有する点の集合である点群データと呼ばれるデータの収集が実用化している。しかし、点群データは実務における様々な意思決定において未だ十分に活用されているとはいえ、課題の 1 つとして、地物ごとの変状や時系列変化などを容易に把握できないことがあげられる。地物の変状に関しては、実際に現地に変状が生じたという情報が判明した際に変状前後の状況を事後的に比較するような問題にしか活用できていないのが現状である。このような背景から、大規模点群を処理可能な汎用的異常検知技術の開発がさらなる点群データの活用に資すると考えられる。また、人材や予算の制約により、近接目

視によらず社会基盤施設の変状の有無を確認可能な技術が求められている<sup>21)</sup>。現状では、本研究で対象とする土工構造物の点検は近接目視点検によって行われている。これは切土・盛土構造物や地形条件・変状形態などの一般事項の把握、点検対象の机上確認、徒歩による現場目視点検、膨大な点検箇所の記録というフローに従って実施されている。しかし、点検者が地形条件や着目点を理解している必要があるため暗黙知に頼る面が強いことや、近年増加傾向にある高盛土 (盛土高 10m 程度以上<sup>22)</sup>) や長大切土 (切土高 15m 程度以上<sup>22)</sup>) の目視点検に多大な労力を要することなどの問題点があげられている。近接目視点検の中には、ひび割れ幅の計測や傾斜量の計測、移動量・沈下量の計測、はらみだしの計測などが含まれており、これらの変状を自動で簡易に検知することができれば業務の効率化につながると考えられる。

### (2) 異常検知

異常検知は、一般的に観測データが正常か異常かを判定するためのラベルがデータに付されているかどうかで 3 つの手法に分類される。観測値に加え、正常/異常のラベルが付されているデータを用いる場合を教師あり異常検知、付されていないデータを用いる場合を教師なし異常検知という。また、正常の方のみラベルが付されている場合など、一方のラベルのみ付されているデータを用いる場合を半教師あり異常検知という。教師あり異常検知では、少数の異常データと正常データに基づき、それらの分類モデルを学習することによって異常の検知を行う。一方、教師なし異常検知では、正常データのみを学習させることによって正常データのみの特徴を表すモデルを作成し、新しいデータを入力した際の異常度を計算することにより異常の検知を行う。このような分類のもと、異常検知を社会基盤施設に適用する場合、教師あり異常検知ではいくつかの問題が生じることが多い。第一に、社会基盤施設は一般に正常状態であることが求められ、異常が発生することは稀であるため、異常データの収集が難しい。第二に、正常/異常のラベル付けは人手で行われることが通常であるため、ラベル付きデータの収集が難しいということがあげられる。このような背景から、社会基盤施設における異常検知では教師なし異常検知が有用であると考え、本研究においては教師なし異常検知手法を採用する。

従来より教師なし異常検知は様々な分野で研究がなされており、数多くの手法が提案されてきた。古典的かつ統計的な異常検知は、観測データの母集団が従う確率分布を仮定し、母数を推定することによってデータにモデルをフィッティングさせ、この分布に基づき新

規入力の異常度を算出するというステップで構成される。この際、正常データは確率の高い領域で発生し、異常データは確率の低い領域で発生すると仮定している。代表的な手法として、スミルノフ・グラブス検定や混合ガウス分布に基づく手法、ホテリングの  $T^2$  理論などがあげられる<sup>23)</sup>。これらの手法では、データが高次元になるとモデルをフィッティングさせることが難しくなる問題点がある。

古典的かつ統計的な異常検知に対して、ルールベースによる異常検知は、正常時のデータにおける汎用的ルールを事前に設定しておき、それに従わないデータが生じたときに異常と判断する手法である。この手法は、専門家のドメイン知識を直接的に導入可能であることや、既知異常によく合致するルールを設定できることが特徴である。ルール設定に関する代表的な手法には、RIPPER<sup>24)</sup>などがあげられる。しかし、ルールベースによる異常検知では未知異常に対応するルールを作成することは難しく、ルールが複雑化しすぎた場合には専門技術者であってもルール設定が困難になるという問題が存在する。

機械学習による異常検知は近年主流となっており、クラスタリングや最近傍法、分類器によるものなど様々な種類が含まれる。具体的な手法としては、One-class Support Vector Machine や Local Outlier Factor (LOF) など<sup>23)</sup>があげられる。機械学習による異常検知は、一般にはどのような分布を有するデータに対しても有効であるが、異常検知精度がデータの質に強く依存する。なお、このほかにも様々な分類が存在し、本分類において必ずしも明確に判別する基準は存在しないことに注意されたい。

### (3) 深層学習と異常検知

上記で述べた高次元に起因する課題などを解決するため、近年、機械学習の中でも深層学習に基づく異常検知手法の研究蓄積が著しい。深層学習は特徴抽出自体を学習することができ、定義が難しい特徴（異常データの基準）を考慮しやすい。また、画像データなどにおいて高次元な特徴を抽出する場合に極めて有効である。深層学習に基づく異常検知手法はいくつも研究がなされているが、それらは AutoEncoder による異常検知<sup>25),26)</sup>、敵対的生成ネットワーク (Generative Adversarial Nets, 以下 GAN)<sup>27),28)</sup> による異常検知、ハイブリッドモデル (深層学習の特徴抽出+異常検知モデルのアプローチ)<sup>29),30)</sup> による異常検知などに大別される。

AutoEncoder については、3.(3) で詳細を述べるため、ここでは概要のみを述べる。図-1 には、AutoEncoder による異常検知の概念を示すので、併せて参照されたい。AutoEncoder は入力と同じデータを出力として復元す

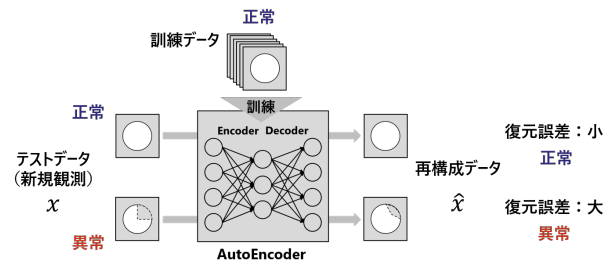


図-1 AutoEncoder による異常検知

るような AI である。AutoEncoder による異常検知では、はじめに正常データのみを用いて AutoEncoder を学習させる。その後、判定したい新規データを学習済みの AutoEncoder に入力し、入力と復元との復元誤差を算出する。このとき、新規データが正常なものであれば、復元も正常なものであるため復元誤差が小さくなる。一方で、新規データが異常なものであれば、AutoEncoder は正常データの特徴しか学習していないため、異常データを復元することができず、復元誤差が大きくなる。したがって、復元誤差を確認することによって、入力が正常であったか異常であったかを判別することが可能である。

GAN による異常検知も AutoEncoder に類似した方法を用いる。GAN は生成器と識別器という2つのニューラルネットワークから構成されている。生成器は、識別器に本物と識別させるような実データに近いデータを生成することが目的であり、識別器は生成器による生成データと実データを区別することが目的である。これを利用すれば、正常データのみで学習させた生成器では異常データを生成することができないため、新規入力と生成データの誤差を確認することによって異常を検知可能である。

ハイブリッドモデルによる方法では、まず適切なニューラルネットワークを用いて特徴抽出を行い、高次元な実データの次元を削減する。続いて、これを従来の統計モデルや機械学習モデルなどの入力として用いることによって異常検知を行う。そのほかにも、時系列データに関する異常検知では長短期記憶 (Long Short Term Memory, 以下 LSTM) などのモデルが用いられることもある。本研究では、教師なし異常検知手法の中で最も広く用いられるモデルの1つであり、扱いが簡便であることと 2.(5) に示す理由から AutoEncoder による異常検知手法を用いる。

### (4) 点群の特性

点群データとは、空間座標  $(x, y, z)$  や色情報  $(R, G, B)$  などを有する、レーザーにより取得される点の集合である。近年、レーザー計測技術の発展により点群データ

の取得が容易化し、様々な分野で活用の試みが進んでいる。点群は移動計測車両測量システム (Mobile Mapping System, 以下 MMS), 航空 LP (Laser Profiler), 無人航空機 (Unmanned Aerial Vehicle, 以下 UAV), 地上設置型レーザースキャナー, ハンドヘルド LP など様々な方法を用いて計測される。これらによって取得された点群データは、一般に LiDAR (Light Detection and Ranging) 点群と呼ばれる。LiDAR とは、計測範囲にレーザ光を照射し、その反射光が観測されるまでの時間や反射光の反射強度を解析することによって、対象物までの距離や反射特性、座標値を取得する光学リモートセンシング技術である。反射光から反射点の空間的情報が得られ、このような反射点を無数に集めると点群データとなる。本研究では、上記のうち MMS によって取得された点群データに着目して、5. の適用事例において用いる。

MMS は、車両にレーザ装置とデジタルカメラを搭載した計測システムである。GNSS(global navigation satellite systems) アンテナとオドメータ、慣性計測装置 (inertial measurement unit, IMU), 走行距離計 (distance measurement indicator, DMI) などによりレーザ照射時の自己位置や姿勢を計測しながら点群データの取得を行う。各種機器が時間的に同期化されたデータに対して後処理を行うことによって、計測時の観測幾何学性を再現し、点群を生成する。デジタルカメラに関しては、画角の狭い各方向への複数のエリアカメラと全周囲カメラが搭載されていることが一般であり、周辺の画像を取得する機器として目的に応じて使用される。また、レーザースキャナー以外に搭載される機器の組み合わせは様々であり、多様な機種が存在する。

### (5) 点群における異常検知

本研究では、データ数に起因する計算負荷の軽減と、本研究で取り扱う局所的異常の検知可能性を考慮して、法枠ごとの点群に対する異常検知を行うとする。法枠ごとの点群に対する異常検知を考えるにあたり、画像における異常検知手法との類似性を用いる。画像における異常検知技術は、生産ラインにおける検査工程の自動化や製造品質のモニタリングなど、さまざまな分野で用いられている。人手による検査を行う場合、不良品の定義を共有する必要があるが、どこまでを不良品としどこまでを正常品とするか、明確な基準をつくることは難しい場合が多く、複数人で基準を統一することは不可能である。また、点検結果が検査人の状態にも依存する可能性がある。深層学習による異常検知を用いることで、このような個人の特性を排除することが可能である。また、不良品の基準を形や色などから明確に定めることは一般に難しい。深層学習を用い

れば、定義が困難な不良品の特徴を考慮することが可能である。法面における異常もこれと同様であり、後述するようなはらみやねじれなどの条件をすべて列挙して異常を定義することは困難である。

なお、点群においては差分解析により変状を抽出するという手法が一般的である。2 時期の点群では、上空の視界や GPS 衛星の配置状況に起因して、相対的・絶対的な位置ずれが生じることが一般である。このような位置ずれは、場合によっては数十 cm に達することが報告されている。したがって、生の 2 時期データで差分解析を行うことは困難であり、2 時期点群の位置合わせ (レジストレーション) 処理が必要となる。差分解析手法は、ある程度大規模な点群に利用することができるが、そのようにした場合、局所的な変状を抽出することには不向きとなる。すなわち、大規模な点群を同時に扱うと、細かな変状を抽出することは一般に難しくなる。また、法枠ごとに抽出した点群に対し、それぞれ位置合わせを行うことも考えられるが、2 時期の点群で同一の点が取得されることはないことや、法枠内点群の抽出は人手で行うため 2 時期の点群で同じ抽出を行うことは不可能であることから、たとえ位置合わせを行ったとしても、異なる点が取得されることに起因した誤差により、誤検知が増加する恐れがある。加えて、予め異常が発生する箇所が既知の場合は、該当する法枠のみモニタリングを行えばよいが、それらが未知である実際の状況では、全ての法枠内点群の位置合わせを行うこととなり、法枠の数が増えると煩雑となってしまふ。このような考えの下、本研究では局所的異常検知技術の開発を目的として、AutoEncoder による異常検知を用いることとした。また、AutoEncoder による異常検知手法を用いることで、必ずしも 2 時期の点群ではないような点群データに対しても手法を適用することができる。本研究で対象とする法面点群データの場合には 1 期目のデータに対し手法を適用することで、初期異常や施工不良を可視化することができる。これについての詳細は 5. にて述べる。

## 3. 深層学習の要素技術

### (1) 順伝播型ニューラルネットワーク

#### a) 順伝播型ニューラルネットワークの基本構造

順伝播型ニューラルネットワーク (Feedforward Neural Network), あるいは多層パーセプトロン (Multi-Layer Perceptron) は、高度な情報処理の実現のために、生物の神経回路網を再現可能な仕組みとして開発されたネットワークモデルである。特に、順伝播型ニューラルネットワークの層数が一定数以上であるものは深層ニューラルネットワークと呼ばれ、これを機械学習の学

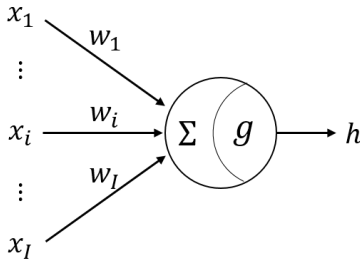


図-2 ニューロンの数理モデル.  $\Sigma$  は各  $x_i w_i$  の  $i$  に関する和をとることを意味する.

習器として用いる手法の総称を深層学習 (Deep Learning) という.

順伝播型ニューラルネットワークは, 図-2 に示すように  $I$  個の要素で構成されるベクトル  $\mathbf{x} = [x_1, \dots, x_I]^T$  を入力とし, スカラー値  $h$  を出力するユニット (ニューロン) とその関係性を表すパラメータ  $\theta$ , および活性化関数  $g$  によって,

$$h = f(\mathbf{x}; \theta) = g(\mathbf{w}^T \mathbf{x} + b) \quad (1)$$

と記述できる. ここで,  $f$  はユニットにおける演算パラメータ  $\theta$  を有し,  $\mathbf{x}$  を引数とする関数として定義される.  $\theta$  は  $\mathbf{w} = [w_1, \dots, w_I]^T \in \mathbb{R}^I$  と  $b \in \mathbb{R}$  で構成され, それぞれ重みベクトル, バイアスパラメータと呼ぶ. また,  $^T$  はベクトル・行列に対する転置操作を表す. 活性化関数は任意の広義単調増加な非線形関数であるが, 通常は双曲線正接関数  $g(z) = \tanh(z)$  やロジスティックシグモイド  $g(z) = 1/(1 + e^{-z})$ , 正規化線形関数 (Rectified Linear Unit, ReLU, あるいはランプ関数)  $g(z) = \max\{0, z\}$  などが用いられる. これによって, 重みやバイアスによる線形変換では捉えられないようなデータの情報を抽出することが可能となる.

議論を一般化するために, 総数  $J$  のユニットを考える. いまユニット  $j$  ( $j = 1, \dots, J$ ) の重み  $w_j$  によりスカラー値  $h_j$  が出力されるとすれば, ユニット  $j$  における入出力の関係式は,

$$h_j = g(\mathbf{w}_j^T \mathbf{x} + b_j) \quad (2)$$

と表される. これは複数のユニットの出力  $h_j$  をベクトル  $\mathbf{h} = [h_1, \dots, h_J]^T$ , 重みを  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_J]$ , バイアスを  $\mathbf{b} = [b_1, \dots, b_J]^T$  と表すことにより,

$$\mathbf{h} = g(\mathbf{W}^T \mathbf{x} + \mathbf{b}) \quad (3)$$

と記述することができる. 順伝播型ニューラルネットワークでは式 (3) をネットワークの層と呼ぶ.

順伝播型ニューラルネットワークは, この層を重ねることによって構成され,  $l$  層目の出力, パラメータ, 活性化関数をそれぞれ  $\mathbf{h}^{(l)}$ ,  $\theta^{(l)} = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}$ ,  $g^{(l)}$  と

すれば,

$$\begin{aligned} \mathbf{h}^{(l)} &= f^{(l)}(\mathbf{h}^{(l-1)}; \theta^{(l)}) \\ &= g^{(l)}(\mathbf{W}^{(l)T} \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}) \\ &= g^{(l)}(\mathbf{W}^{(l)T} g^{(l-1)}(\mathbf{W}^{(l-1)T} \mathbf{h}^{(l-2)} \\ &\quad + \mathbf{b}^{(l-1)}) + \mathbf{b}^{(l)}) \end{aligned} \quad (4)$$

と再帰的に表され,  $l$  層目の出力はそれ以前のすべての層のパラメータに依存する. ネットワークが合計  $L$  層で構成される場合, 最終層 ( $L$  層) の出力を  $\hat{\mathbf{y}} = \mathbf{h}^{(L)}$  と書く. また,  $f^{(1)}$  を入力層,  $f^{(L)}$  を出力層, それ以外の中間層  $f^{(2)}, \dots, f^{(L-1)}$  を隠れ層と呼ぶ.  $L$  層を有するニューラルネットワークの構造を図-3 に示す. なお, 層数に関しては, ユニット間の接続を層と呼ぶ場合と, ユニット自体を層と呼ぶ場合がある. 本研究では前者を採用している.

## b) 順伝播型ニューラルネットワークの性質

ニューラルネットワークにおける学習 (訓練) について説明する. 入力  $\mathbf{x}$  とそれに対応する目標出力  $\mathbf{y}$  の真のデータ分布  $p^*(\mathbf{x}, \mathbf{y})$  を考える. 学習の目的は真のデータ分布のもとで,  $\mathbf{x}$  を入力としたときの出力  $\hat{\mathbf{y}}(\mathbf{x}; \theta)$  を目標出力  $\mathbf{y}$  に近づけるようにパラメータ  $\theta$  を調整することである. これは, 出力  $\hat{\mathbf{y}}$  と目標出力  $\mathbf{y}$  の誤差をある関数  $j(\hat{\mathbf{y}}(\mathbf{x}; \theta), \mathbf{y})$  で表し, これを  $\theta$  について最小化することによって実現できる. この関数を誤差関数や損失関数, 目的関数などと呼ぶ. つまり, 求めたい最適な  $\theta$  の値を  $\theta^*$  とおくと, 学習は最適化問題

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim p^*(\mathbf{x}, \mathbf{y})} [j(\hat{\mathbf{y}}(\mathbf{x}; \theta), \mathbf{y})] \quad (5)$$

として表現できる. 実際には真のデータ分布  $p^*(\mathbf{x}, \mathbf{y})$  は未知であるので, 式 (5) を直接計算することは不可能である. そこで, 真のデータ分布からのサンプル集合  $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$  を用いて式 (5) を

$$\theta^* = \arg \min_{\theta} \sum_{n=1}^N j(\hat{\mathbf{y}}(\mathbf{x}_n; \theta), \mathbf{y}_n) \quad (6)$$

によって近似計算する. 真のデータ分布からのサンプル集合  $\mathbb{D}_{\text{train}} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$  を訓練集合 (訓練データ) といい,  $J(\theta) = \sum_{n=1}^N j(\hat{\mathbf{y}}(\mathbf{x}_n; \theta), \mathbf{y}_n)$  を訓練誤差と呼ぶ. 本来の学習の目的は, 式 (5) の最適化であったが, 式 (6) に近似して計算を行うため, パラメータ  $\theta^*$  として訓練集合にのみ適合するものが選定されることがある. この現象を過学習 (過剰適合) という. 過学習が生じると, 訓練集合には現れていない将来予測したいサンプルに関する誤差 (汎化誤差) と訓練誤差にギャップが生じてしまう. 本来評価したいものは汎化誤差であるため, 過学習を防ぐことが必要となる. そこで, 訓練集合とは別のサンプル集合  $\mathbb{D}_{\text{test}} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_M, \mathbf{y}_M)\}$  を用意しておき, モデルの学習の終了後にこのサンプルに関する誤差

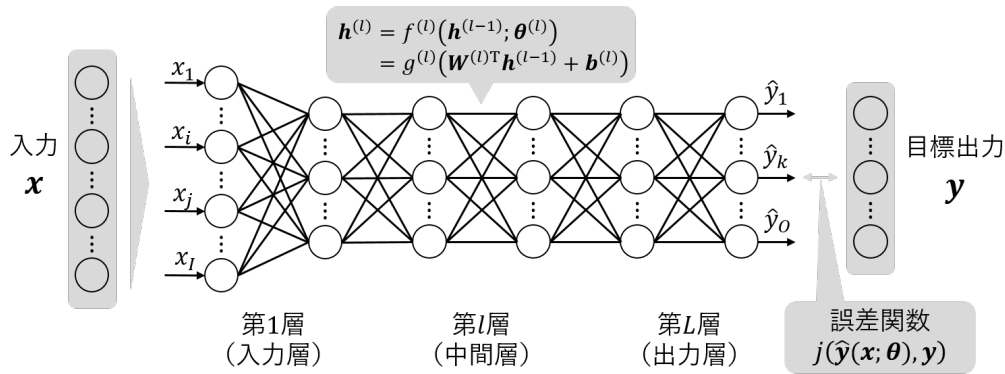


図-3 順伝播型ニューラルネットワークの構造

$\sum_{m=1}^M j(\hat{\mathbf{y}}(\mathbf{x}_m; \boldsymbol{\theta}), \mathbf{y}_m)$  を評価することにより過学習の程度を確認すればよい.  $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_M, \mathbf{y}_M)\}$  をテスト集合 (テストデータ),  $\sum_{m=1}^M j(\hat{\mathbf{y}}(\mathbf{x}_m; \boldsymbol{\theta}), \mathbf{y}_m)$  をテスト誤差と呼び, これにより汎化誤差の評価を行う.

以上のように訓練誤差を最小化することによってモデルを学習するが, 訓練誤差は一般にパラメータに関して凸関数にならず, 大域的な最小解を解析的に求めるのは困難である. そのため, 特に層の深いネットワークにおいては, 大域的な最小解の代わりに局所的な極小解を求めることを考える. 深層学習において極小解を得るために用いられる最も代表的な手法として, 勾配降下法

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \frac{\partial J(\boldsymbol{\theta}^{(t)})}{\partial \boldsymbol{\theta}^{(t)}} \quad (7)$$

があげられる. ここで  $\eta$  を学習率 (学習係数) と呼び, 1 回あたりの更新幅を意味する. また,  $t = 1, 2, \dots$  は時間ステップを表す. 式 (7) ではすべてのパラメータに関する勾配を計算する必要があるが, 訓練誤差は出力層において計算されているため, 隠れ層における勾配を計算することが不可能である. そこで, 合成関数の微分法 (連鎖律) を利用して, 出力層における誤差を入力層の方向に伝播させる誤差逆伝播法により隠れ層の勾配を計算する.

式 (7) では, 全訓練データに対する誤差を用いてパラメータを更新しているが, 実際には訓練データの量が増加するにつれて計算を行うことが困難となる. そこで, 訓練集合の中から, ステップごとにランダムに部分集合 (ミニバッチ) を取り出し, そのデータで計算した勾配に基づきパラメータを更新する. この方法を確率的勾配降下法 (stochastic gradient decent, SGD) という. SGD において, 1 つの訓練集合を繰り返し学習させる回数をエポック数と呼ぶ. 以上においては, データとして  $(\mathbf{x}, \mathbf{y})$  というペアが与えられた場合を想定した. これを  $\mathbf{x}$  に対応するラベル  $\mathbf{y}$  が与えられているという意味で教師あり学習と呼ぶ. 一方で, データとし

て  $\mathbf{x}$  のみが与えられる枠組みを教師なし学習と呼ぶ.

ニューラルネットワークにおいて, 重みの初期値は学習の進み方や精度に大きく影響する. すなわち, 偏ったデータが各層を流れると勾配が消失したり, 表現力が制限されたりするため, 学習が困難となる. そこで, 各層の出力に適度な広がりをもたせるため, Xavier<sup>31)</sup> や He<sup>32)</sup> と呼ばれる初期化法が用いられる. Xavier の初期値においては, 前層のユニット数を  $n$  としたとき平均 0, 標準偏差  $\frac{1}{\sqrt{n}}$  の正規分布  $\mathcal{N}(0, \frac{1}{\sqrt{n}})$  から重みを抽出する. He の初期値においては  $\mathcal{N}(0, \sqrt{\frac{2}{n}})$  から重みを抽出する. 一般に, 活性化関数に左右対称な S 字曲線 (tanh, ロジスティックシグモイドなど) を用いる場合は Xavier の初期値が適しており, ReLU を用いる場合は He の初期値が適しているといわれている. これは,  $\text{ReLU}(z) = \max\{0, z\}$  の場合は負値の入力に対しすべて出力が 0 となるため, 正規分布の標準偏差を大きくし, より出力に広がりをもたせる必要があるためである. また, 各層で適度な広がりをもつように強制的に分布を調整する手法として, バッチ正規化 (batch normalization)<sup>33)</sup> という手法が提案されている. これは, 訓練時に各層の出力 (ミニバッチ) に対し標準化 (データ集合の平均を 0, 分散を 1 にする) を行うものであり広く用いられている.

## (2) 畳み込みニューラルネットワーク

### a) 畳み込みニューラルネットワークの性質

画像などの高次元データを扱う場合, 順伝播型ニューラルネットワークではいくつかの問題が生じる. 順伝播型ニューラルネットワークでは, 重み行列の積により層間ユニットの結合が定義されており, 層間の全ユニットが結合状態となっている. そのため, 画像などの高次元データに対し順伝播型ニューラルネットワークを用いると, 重みの数が膨大となり学習が困難となる. また, 全結合層への入力では高次元のデータを 1 次元にリシェイプする必要があり, データの形状が無視されて

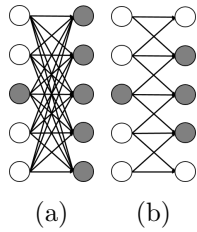


図-4 層間結合の違い。(a) 全結合層, (b) 畳み込み層

しまう。例えばカラー画像の場合、縦、横、チャンネル方向の3次元情報を有するが、これを1次元に変換して入力する場合、ピクセル間の空間的自己相関やチャンネル間の関係などの情報を喪失してしまう。この点に関して、畳み込みニューラルネットワーク (Convolutional Neural Network, 以下 CNN)<sup>34)</sup> は有効な性質を有している<sup>35)</sup>。CNN は格子状のトポロジーを持つデータの処理に使われるニューラルネットワークである<sup>35)</sup>。このようなデータの例としては、時系列データ (等間隔で取得したサンプルが1次元に配列されたもの) や画像データ (ピクセルが2次元に配列されたもの) があげられる。CNN では通常のニューラルネットワークと異なり、少なくとも1つの層で畳み込みと呼ばれる処理が用いられる。

CNN の性質について、第一に、局所性や局所受容野と呼ばれるように、層間ユニットの結合が疎であることがあげられる。CNN では入力より小さなフィルタを用いることで、局所的な結合で処理を行うことができ、有意な特徴を検出することが可能である。これにより、空間的自己相関に配慮しつつも大幅なパラメータ数の削減が可能である。このような場合でも、深い層のユニットにおける受容野 (あるユニットに結合するすべての上位ユニット) は浅い層におけるものより大きくなるため、CNN では直接的な結合が疎であっても、深い層のユニットは入力のほぼすべてと間接的に結合している。図-4 に順伝播型ニューラルネットワークと CNN の層間結合の違いを示す。

第二に、重み共有 (パラメータ共有) と呼ばれるように、結合の重みが出力マップのチャンネルに関し共有されることがあげられる。これは、画像のある位置で重要な特徴は他の位置でも重要である可能性が高いという仮定に基づいており、さらなるパラメータの削減に寄与している。すなわち、チャンネル  $m$  において  $u_{ijm}$  の各画素に対し、同一の重み  $h_{pqkm}$  が共有される。このような重み共有により、畳み込み層は並進不変性 (移動不変性) を得ることになる。すなわち、入力が少し並進移動したとしても出力がほぼ不変であるという性質

を得る。なお、回転やスケーリング (拡大・縮小) に対しては不変性はないので、データ拡張によりそのようなデータを増やして学習するなどの対応が必要である。

## b) 畳み込みニューラルネットワークの基本構造

離散値で定義された関数に対する畳み込みは、

$$(x * w)(p) = \sum_q x(q)w(p-q) \quad (8)$$

で与えられる。CNN においては、第一引数  $x$  を入力、第二引数  $w$  をフィルタ (カーネル)、出力  $x * w$  を特徴マップと呼ぶ。一般に、入力  $x$  は多次元の配列である。典型的な例として、画素  $x_{ij} \in \mathbb{R}$  をもつ  $W \times W$  の2次元画像 (グレースケール画像)  $\mathbf{X} = [x_{ij}]$  における畳み込みは、 $H \times H$  の2次元フィルタ  $\mathbf{H} = [h_{pq}]$  を用いて、

$$\begin{aligned} u_{ij} &= \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{pq} h_{i-p, j-q} \\ &= \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i-p, j-q} h_{pq} \end{aligned} \quad (9)$$

と表せる。ここで、式 (8) における引数を式 (9) では添え字として表し、畳み込みの性質である可換性を用いた。ニューラルネットワークの実装においては、慣用的にフィルタを上下左右に反転したものである相互相関関数と呼ばれる次の式を畳み込みとして定義しており、以後本研究においても、

$$u_{ij} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p, j+q} h_{pq} \quad (10)$$

を畳み込みと称する。カラー画像のような多チャンネル画像では、複数個のフィルタを用意し、並行して畳み込みを行う。多チャンネル画像とは、各画素が  $K (> 1)$  個の値を有する場合の画像であり、RGB 画像の場合  $K = 3$  である。RGB 画像における畳み込みは、画素  $x_{ijk} \in \mathbb{R}$  を有する  $W \times W \times K$  の3次元画像と、画素  $h_{pqk} \in \mathbb{R}$  を有する  $H \times H \times K$  の3次元フィルタを用いて、各チャンネルについて並行に画像とフィルタの畳み込みを行った後、全チャンネルについて各画素ごとに総和

$$u_{ijm} = \sum_{k=0}^{K-1} \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p, j+q, k} h_{pqkm} + b_{ijm} \quad (11)$$

をとる。ここで、フィルタ数 (=出力マップのチャンネル数) を  $M$  とし、 $m = 0, \dots, M-1$  である。また、式 (11) ではバイアスを導入している。一般に  $b_{ijm} = b_m$  とフィルタごとに共通とすることにより、画像全体の濃淡を表すようにする。

典型的な CNN では、畳み込みの後にプーリングと呼ばれる処理が行われる。プーリングとは、ある場所での出力を周辺出力の要約特徴量で置き換える処理のことである。この要約処理によって、入力の小さな並進移動に対しプーリングの値はほぼ不変となり、入力に対す

る並進不変性の向上につながる。サイズ  $W \times W \times K$  の入力画像上で画素  $(i, j)$  を中心とする  $H \times H$  の矩形領域内の格子点集合  $Q_{ij}$  をとったとき、 $L_P$  プーリングとは、

$$u_{ijk} = \left( \frac{1}{H^2} \sum_{(p,q) \in Q_{ij}} z_{pqk}^P \right)^{\frac{1}{P}} \quad (12)$$

で表される処理のことである。プーリングの計算は各チャンネルで独立に行われていることに注意を要する。一般に、 $P = 1$  の場合である Average プーリングや、 $P = \infty$  の場合である Max プーリングが広く用いられる。プーリング層においては活性化関数を適用しないことが通常であり、学習パラメータも存在しない。このほかに、局所コントラスト正規化層 (Local Contrast Normalization, 以下 LCN) <sup>36)</sup> などを併用することがある。

### (3) AutoEncoder

AutoEncoder(自己符号化器)<sup>37)</sup> とは目標出力をもたず、入力のみで訓練データを用いた教師なし学習であり、データを良く表す特徴表現の獲得を目的としたニューラルネットワークである。良い表現については、明確に定められた定義は存在しないが、一般に情報量や独立性、説明性、スパース性、不変性、滑らかさなどを有するようものであるといわれている<sup>38)</sup>。提案当初は深いネットワークの事前学習(重みの良い初期値を得ること)に用いられていたが、近年は次元削減や異常検知、ノイズの除去などに利用されることが多くなっている。

入力ベクトル  $\mathbf{x} = [x_1, \dots, x_I]^T$  を受け取り、出力  $\mathbf{h} = [h_1, \dots, h_H]^T$  をもつ単層のニューラルネットワークを考える。すなわち、

$$\mathbf{h} = f_e(\mathbf{x}; \boldsymbol{\theta}_e) = g_e(\mathbf{W}_e^T \mathbf{x} + \mathbf{b}_e) \quad (13)$$

を考える。このネットワークを出力層で折り返せば、図-5 に示すように入力層と出力層に同数のユニットをもつ 2 層のニューラルネットワークとなる。折り返した層の活性化関数を  $g_d$ 、重みを  $\mathbf{W}_d$ 、バイアスを  $\mathbf{b}_d$  とすれば、この層における出力  $\hat{\mathbf{x}} = [\hat{x}_1, \dots, \hat{x}_I]^T$  は、

$$\hat{\mathbf{x}} = f_d(\mathbf{h}; \boldsymbol{\theta}_d) = g_d(\mathbf{W}_d^T \mathbf{h} + \mathbf{b}_d) \quad (14)$$

で表される。中間出力  $\mathbf{h}$  を入力  $\mathbf{x}$  の符号 (code) とみなし、 $f_e$  を符号化 (encode)、 $f_d$  を復号化 (decode) と呼ぶ。なお、ここでは 2 層の AutoEncoder を示したが、多層に拡張することは可能である。AutoEncoder において、符号化する層全体をエンコーダ、復号化する層全体をデコーダという。

AutoEncoder における学習の目的は入力  $\mathbf{x}$  を符号化し、再度復号化したとき、再構成されたデータ  $\hat{\mathbf{x}}$  が極力もとの入力  $\mathbf{x}$  に類似するように符号化すること、す

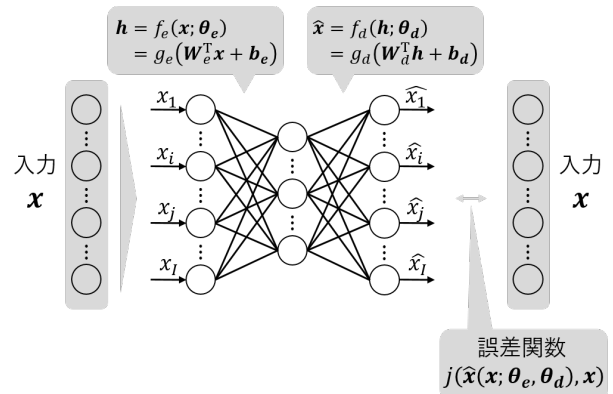


図-5 AutoEncoder の構造

なわち重みを定めることである。したがって、学習における誤差関数に関しては、入力  $\mathbf{x}$  に応じて、 $\mathbf{x}$  と  $\hat{\mathbf{x}}$  の非類似度を表す指標を用いる。 $x_i \in \mathbb{R}$  の場合、 $g_d$  を恒等写像とし、誤差関数は、

$$J(\boldsymbol{\theta}) = \sum_{n=1}^N \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2 \quad (15)$$

で表される二乗和誤差となる。 $x_i \in \{0, 1\}$  の場合 (独立に 2 値を取る場合)、 $g_d$  をロジスティックシグモイドとし、誤差関数は、

$$J(\boldsymbol{\theta}) = - \sum_{n=1}^N \sum_{i=1}^I [x_i \ln \hat{x}_i + (1 - x_i) \ln (1 - \hat{x}_i)] \quad (16)$$

により表される交差エントロピー誤差となる。なお、中間層の活性化関数  $g_e$  は自由に定められ、通常は非線形関数を選択する。出力層の活性化関数  $g_d$  は、上記のように目標出力に応じて定める。最も単純な場合として  $g_e, g_d$  がともに恒等写像の場合を考えれば、ネットワーク全体が表す関数は、

$$\hat{\mathbf{x}} = \mathbf{W}_d(\mathbf{W}_e \mathbf{x} + \mathbf{b}_e) + \mathbf{b}_d \quad (17)$$

のような線形写像で表される。このとき、 $H \leq I$  (不完備と呼ぶ) であれば誤差関数を最小にする  $\mathbf{W}_e$  および  $\mathbf{W}_d$  は訓練データ  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  に対する主成分分析により得られるものと等しくなる。すなわち、AutoEncoder とは主成分分析の非線形への拡張である。

### (4) 点群深層学習

点群深層学習とは、3次元点群を深層学習で扱う枠組みの総称を指し、PointNet<sup>9)</sup> と呼ばれるネットワークの提案以降、急速に発展を遂げている技術である。従来の点群に対する深層学習技術では、深層学習が画像認識分野で大きな成功を収めたことから、点群をボクセル (複数点群を一纏まりにしたもの) に変換し、3次元畳み込みニューラルネットワークに入力する手法が研究されていた<sup>39),40)</sup>。しかし、点群をボクセルに変換する際に点群個々の位置情報が失われたり、点群の空



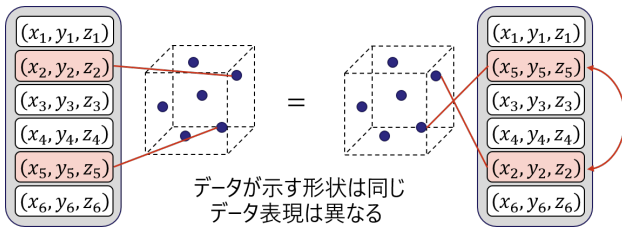


図-6 点群データにおける順不変性

間特性が必然的に変化するため物体の凹凸表現が失われたりする問題点があった。

このような問題点を踏まえ、点群を他の形式に変換せず、点のまま入力する手法の研究が試みられた。そのような手法を実現するためには、図-6 のようにデータの順番に左右されないモデルの構築（順不変性の考慮）が不可欠となる。すなわち、画像データの場合、グリッドに固定された構造をもつため順序が一意に決定されるのに対し、点群データをそのままの形式で扱う場合やグラフなどは入力の順序を任意に決められるため、順不変性を考慮する必要が生じる。考えられるアプローチの例として、1) 一定の規則に従い頂点をソートし直す方法、2) データ拡張により頂点の順番に影響を受けにくいモデルを作成する方法、3) 対称関数とよばれる変数の順番を入れ替えても値が不変な関数を学習する方法などが存在する。これらのうち、1) では高次元空間での並び替えは困難であるという問題点がある。また、2) では頂点が  $N$  個ある場合の順列は  $N!$  であるため、データの数が増加して爆発的に増加してしまう。3) における対称関数は数式で表現すれば次のようになり、いわゆる対称式の定義を整式から関数へ拡張した

$$f(\sigma(x_1), \dots, \sigma(x_n)) = f(x_1, \dots, x_n) \quad (18)$$

で与えられる。ただし、 $\sigma$  は  $n$  次対称群  $\mathfrak{S}_n$  の任意の元、すなわち置換である。上記の性質を満たす関数は集合を引数にとる関数として簡潔に  $f(\{x_1, \dots, x_n\})$  と表現することができる。対称関数の例としては、

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n x_i \quad (19)$$

$$f(x_1, x_2, \dots, x_n) = \max\{x_1, x_2, \dots, x_n\} \quad (20)$$

があげられる。PointNet は上記のうち 3) のアプローチを採用した手法である。PointNet では対称関数を

$$f(\{x_1, \dots, x_n\}) \approx g(\mathbf{h}(x_1), \dots, \mathbf{h}(x_n)) \quad (21)$$

のように学習することによって、ネットワークにおける順不変性を獲得する。ここで、 $f: \mathcal{P}(\mathbb{R}^N) \rightarrow \mathbb{R}$ 、 $h: \mathbb{R}^N \rightarrow \mathbb{R}^K$ 、 $g: \underbrace{\mathbb{R}^K \times \dots \times \mathbb{R}^K}_n \rightarrow \mathbb{R}$  は対称関数である。集合  $A$  に対し、 $\mathcal{P}(A)$  は  $A$  の冪集合を表すとする。 $N$  は点群のもつ情報の次元であり、座標情報のみを用い

る場合は  $N = 3$  である。 $K$  は変換後の点群の次元である。PointNet においては、 $h$  を重みを共有させた MLP、 $g$  を 3.(2) で述べた Max プーリングと 1 変数関数（これも MLP を用いている）の合成でモデル化している。すなわち、PointNet における処理は、重みを共有させた MLP により点ごとに特徴を抽出した後、それらを Max プーリングにより集約させるという 2 つの処理から構成される。点ごとの特徴抽出は、入力の順序によらず、全体から集約する操作も順序によらないため、この合成写像も順序によらない。

PointNet 全体の構造は図-7 に示す通りとなっており、

$$f(\{x_1, \dots, x_n\}) = \gamma \left( \text{MAX}_{x_i \in S} \{h(x_i)\} \right) \quad (22)$$

と表せる。ここで、 $S = \{x_i \in \mathbb{R}^N | i = 1, \dots, n\}$  は点群であり、 $\gamma$ 、 $h$  は多層パーセプトロンである。MAX は Max プーリングを表し、 $n$  個のベクトル  $h(x_i)$  に対し、要素ごとに最大値を取得するものとする。図-7 における MLP の括弧内の数字は層数を表す。すなわち、初期の MLP では  $(n, 3) \rightarrow (n, 64) \rightarrow (n, 64)$  と変形される。

PointNet はクラス分類やセグメンテーションなどのタスクに活用されるモデルであり、そのようなタスクでは剛体変換に対する不変性が有用となる。そのため、PointNet では T-Net と呼ばれる座標変換行列を学習するミニネットワークを構造に組み込むことで、剛体変換に対する不変性を獲得している。なお、図-7 において、 $k$  は分類クラス数である。

PointNet 以降、点群を扱う深層学習の手法が盛んに研究されている。PointNet++<sup>10)</sup> では、PointNet を階層的に適用することにより、PointNet が考慮していなかった点群各点の周辺情報に着目している。Xu 等<sup>11)</sup> は、点と点の距離や角度をもとに重みを算出し畳み込みを行う SpiderCNN を提案している。また、これらの認識タスクのほかに 3 次元形状の生成に関する研究も行われている。Yang 等<sup>12)</sup> は、折りたたみを多層パーセプトロンでモデル化し、点群を生成するモデルを提案している。Dai 等<sup>13)</sup> は、アテンション機構などを用いて人間の複雑な形状を生成するネットワークを提案している。Zhang 等<sup>14)</sup> は、点群をアップサンプリングして出力する生成ネットワークを提案している。本研究では、これらの点群生成モデルをもとに異常検知におけるネットワークを構成する。

## 4. 異常検知手法

### (1) PointAutoEncoder

本研究では、AutoEncoder に PointNet における順不変性を獲得する構造を導入することによって、入力として 3 次元点群を扱えるように拡張したモデルを提案

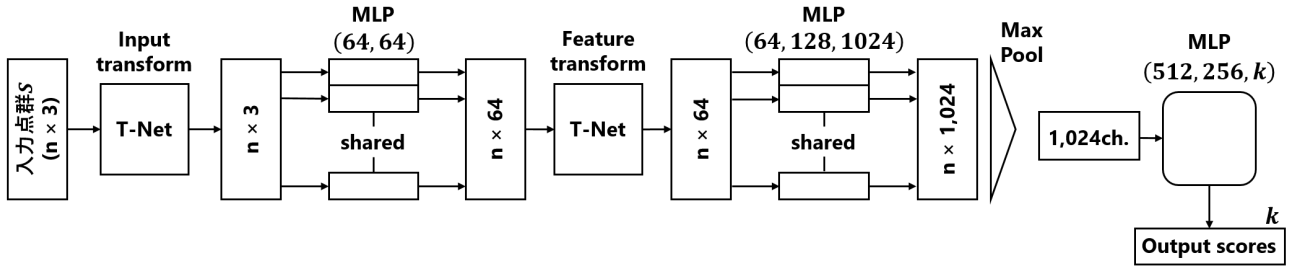


図-7 PointNet の構造. 括弧内の数字は層数を表す.

する. 以下, このモデルを PointAutoEncoder と呼ぶ. はじめに, PointAutoEncoder のエンコーダ部分について説明する. 3.(4) で述べた通り, エンコーダは入力に対する順不変性を有する必要がある. すなわち, 入力の順序を任意に置換したとしてもエンコーダの出力は不変である必要がある. そこで, PointNet における対称関数を学習するネットワーク構造を援用し, エンコーダを

$$\mathbf{h} = f_e(S; \theta_e) = \text{MAX}_{\mathbf{x}_i \in S} \{f_{1 \times 1 \text{Conv}}(\mathbf{x}_i)\} \quad (23)$$

と定義する. ここで,  $1 \times 1 \text{Conv}$  はフィルタサイズ 1 の 1 次元畳み込みニューラルネットワークであり,  $1 \times 1$  畳み込み, pointwise 畳み込みなどと呼ばれる.  $1 \times 1$  畳み込みは 3.(2) で述べたような畳み込み層の性質により, 重みを各点で共有した多層パーセプトロン (MLP) と同等である.  $\mathbf{h}$  は Max プーリング後の特徴量である. 本研究では 4 層の  $1 \times 1$  畳み込み層を用いることとし, すべての層で 3.(1) で述べたバッチ正規化を行う. なお, PointNet で考慮されているような剛体変換に対する不変性は今回のタスクでは考慮する必要がない. デコーダに関しては, 順不変性を損失関数で考慮する指針をとるため, 通常の順伝播型ニューラルネットワーク

$$\hat{S} = f_d(\mathbf{h}; \theta_d) = \text{Reshape}(f_{\text{MLP}}(\mathbf{h})) \quad (24)$$

を採用する. ここで, 本研究においては,  $f_{\text{MLP}}$  は 3 層の全結合層とし, 活性化関数は最初の 2 層で ReLU, 最後の層で恒等写像を用いる. Reshape は全結合層への入力のために 1 次元に変換された特徴量を点群の次元に戻す処理である. また, 最初の 2 層でバッチ正規化を用いた. 図-8 にネットワークの全体構造を示す. 図-8 における各数字は行列の列数を表す.

## (2) 損失関数

損失関数には入力点群と再構成点群の非類似度を設定する必要があり, 通常の二乗和誤差などをそのまま用いることはできない. 適切な損失関数には, 少なくとも 1) 点の位置に関する微分可能性, 2) 計算の容易さ, 3) 点群における少量の外れ値データに関するロバスト性の 3 つの性質が求められる<sup>15)</sup>. また, デコーダには順

不変性を考慮しないため, 損失関数により順不変性を考慮する必要がある. これらを満たす関数として, Chamfer Distance (面取り距離) と Earth Mover's Distance (EMD) がしばしば用いられる. Chamfer Distance は,

$$d_{CD}(S, \hat{S}) = \sum_{\mathbf{x} \in S} \min_{\hat{\mathbf{x}} \in \hat{S}} \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \sum_{\hat{\mathbf{x}} \in \hat{S}} \min_{\mathbf{x} \in S} \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \quad (25)$$

と定義できる. ここで,  $S, \hat{S}$  はそれぞれ入力点群, 再構成点群であり,  $\|\cdot\|_2$  は二乗ノルムを表す. Chamfer Distance は一方の点群の各点に対し, 他方の点群における最近傍点を探索し, それらの間の平方距離を足し合わせる.  $d_{CD}$  は連続関数であり, 区分的に滑らかである. なお,  $d_{CD}$  は距離の三公理のうち, 三角不等式を満たさないため厳密には距離関数とは呼べない. 一方, EMD は,

$$d_{EMD}(S, \hat{S}) = \min_{\phi: S \rightarrow \hat{S}} \sum_{\mathbf{x} \in S} \|\mathbf{x} - \phi(\mathbf{x})\|_2 \quad (26)$$

と定義できる. ここで,  $\phi: S \rightarrow \hat{S}$  は全単射である. EMD は最適化問題の 1 つである輸送問題の考え方に基づいている. 本研究では, 計算の簡易さから損失関数として Chamfer Distance を採用する.

## (3) 重みの最適化手法

重みパラメータの最適化手法について説明する. 本研究では 3.(1) で述べた SGD (確率的勾配降下法) を改良したモデルを用いる. SGD は単に勾配の方向に進むために, 勾配方向が最小値の方向に向いていない場合 (誤差関数の関数形が非等方的である場合), 谷において振動することになる. この現象により, パラメータ空間において非効率な探索を行うことになり, 重みパラメータの最適化に長時間を要する場合がある. このジグザグな動きを改善するものとして, モメンタムという最適化手法が存在する. モメンタムにおけるパラメータの更新式は,

$$\theta^{(t+1)} = \theta^{(t)} - \eta \frac{\partial J(\theta^{(t)})}{\partial \theta^{(t)}} + \alpha(\theta^{(t)} - \theta^{(t-1)}) \quad (27)$$

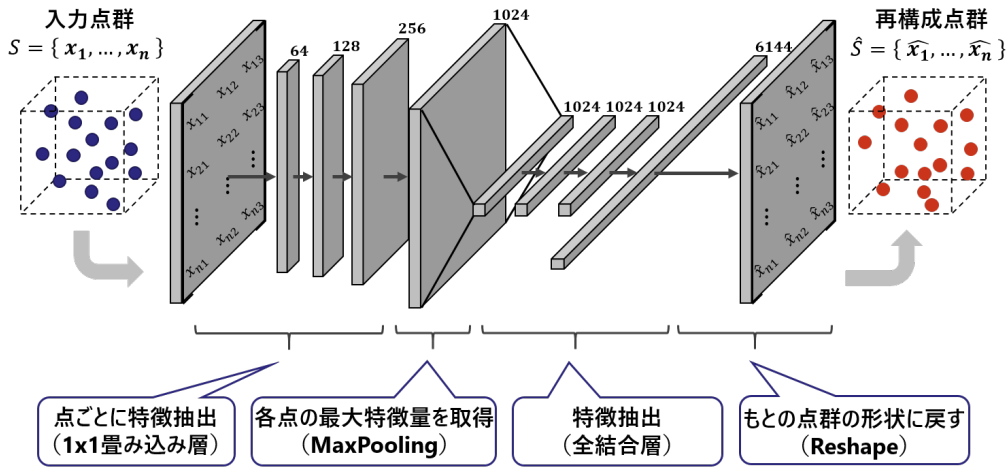


図-8 PointAutoEncoder の構造

と表現される．ここで、 $\alpha \in [0, 1)$  は過去の勾配の寄与が指数関数的に減衰する速度を決定するハイパーパラメータである．式 (27) を変形し、パラメータの更新量を  $\Delta\theta^{(t)} := \theta^{(t+1)} - \theta^{(t)}$  と定義すれば、 $\Delta\theta^{(t)} = \alpha\Delta\theta^{(t-1)} - \eta \frac{\partial J(\theta^{(t)})}{\partial \theta^{(t)}}$  と表現される．すなわち、パラメータの更新量は、過去の更新量と現ステップの値との重み付き平均のような形になる．これにより、谷においてジグザグな動きを示す場合でも、谷直交方向の更新量は自然に消失するため、SGD に比べ早く収束に向かうことができる．

SGD では、学習率  $\eta$  の適切な調整が重要となる．学習率が小さすぎると学習に時間がかかりすぎる一方、大きすぎると発散するため適切に学習を行うことができない．そこで、パラメータの要素ごとに適応的に学習率を調整しながら学習する最適化手法である AdaGrad を用いる．AdaGrad では、勾配の二乗  $\mathbf{h}$

$$\mathbf{h}^{(t)} = \mathbf{h}^{(t-1)} + \frac{\partial J(\theta^{(t)})}{\partial \theta^{(t)}} \odot \frac{\partial J(\theta^{(t)})}{\partial \theta^{(t)}} \quad (28)$$

$$\eta_t = \frac{\eta_0}{\sqrt{\mathbf{h}^{(t)} + \varepsilon}} \quad (29)$$

を蓄積し、学習率の適応的な調整に用いる．式中、 $\eta_0$  は学習率の初期値である． $\varepsilon$  はゼロ除算を防ぐための微小な正定数であり、 $\odot$  は要素ごとの積（アダマール積）を意味する．また、式 (29) における除算や平方根は要素ごとに適用するものとする．この適応的な学習率  $\eta_t$  を利用して、SGD と同様にパラメータ

$$\theta^{(t+1)} = \theta^{(t)} - \eta_t \odot \frac{\partial J(\theta^{(t)})}{\partial \theta^{(t)}} \quad (30)$$

を更新する．このとき、学習率  $\eta_t$  はパラメータの関数となっていることに注意を要する． $\eta_t$  は  $\frac{1}{\sqrt{\mathbf{h}^{(t)}}}$  を含んでおり、勾配が小さい場合学習率が大きくなるため、パラメータの更新量は大きくなる．また、勾配の二乗を蓄積していくため、学習を進めるほど更新幅は小さく

なり 0 に近づく．この現象は、更新幅が 0 に近づいたときに、パラメータが最適点に収束していない場合に問題となる．

そこで、式 (30) における勾配の二乗を保持する式 (28) を

$$\mathbf{h}^{(t)} = \alpha\mathbf{h}^{(t-1)} + (1 - \alpha) \frac{\partial J(\theta^{(t)})}{\partial \theta^{(t)}} \odot \frac{\partial J(\theta^{(t)})}{\partial \theta^{(t)}} \quad (31)$$

のように改善したものを RMSProp という．式 (31) では、過去の勾配の二乗を均一的に加算していくのではなく、ハイパーパラメータ  $\alpha \in [0, 1)$  を導入することによって、徐々に新しい勾配の情報が反映されるようになっていく．時系列の用語を用いれば、式 (31) は勾配の二乗に対する指数（平滑化）移動平均になっており、勾配の二乗値が過去のものから現在にかけて指数関数的に重要視されている．

上記の考え方を組み合わせたアルゴリズムが Adam (Adaptive Moment Estimation) <sup>41)</sup> である．Adam におけるパラメータの更新式は、

$$\mathbf{m}^{(t)} = \beta_1\mathbf{m}^{(t-1)} + (1 - \beta_1) \frac{\partial J(\theta^{(t)})}{\partial \theta^{(t)}} \quad (32)$$

$$\mathbf{v}^{(t)} = \beta_2\mathbf{v}^{(t-1)} + (1 - \beta_2) \frac{\partial J(\theta^{(t)})}{\partial \theta^{(t)}} \odot \frac{\partial J(\theta^{(t)})}{\partial \theta^{(t)}} \quad (33)$$

$$\hat{\mathbf{m}}^{(t)} = \frac{\mathbf{m}^{(t)}}{1 - \beta_1^t} \quad (34)$$

$$\hat{\mathbf{v}}^{(t)} = \frac{\mathbf{v}^{(t)}}{1 - \beta_2^t} \quad (35)$$

$$\theta^{(t+1)} = \theta^{(t)} - \eta \frac{\hat{\mathbf{m}}^{(t)}}{\sqrt{\hat{\mathbf{v}}^{(t)} + \varepsilon}} \quad (36)$$

と表現される． $\beta_1, \beta_2 \in [0, 1)$  はハイパーパラメータであり、指数移動平均の減衰率を意味する．式 (33) は RMSProp における勾配二乗の蓄積と同様な式である． $\mathbf{m}^{(t)}, \mathbf{v}^{(t)}$  はそれぞれ勾配の一次モーメント、二次モーメントと呼ばれる．また、Adam においては RMSProp

と異なり、式 (34)、式 (35) というバイアス補正式が導入されている。これは、一次モーメント、二次モーメントが 0 で初期化されるために、モーメントの推定が 0 に向かいやすいというバイアスを考慮したものであり、特に初期ステップ付近ではバイアスが大きくなりやすい。そのため、通常は  $\beta_1, \beta_2$  を 1 に近い値に設定し、時間が経つにつれ、累指数  $t$  が増加することによって補正率が弱まるようにする。このほかにも、Adadelta など、様々な最適化手法が提案されているが、いずれの最適化手法を用いれば良いかを一概に述べることはできない。本研究では、近年広く用いられていることから Adam を用いることとする。

#### (4) 異常度の定義

本研究では、異常度を点  $x_i \in S$  ごとに算出することとする。点  $x_i$  における各点異常度を

$$R_i = \min_{\hat{x} \in \hat{S}} \|x_i - \hat{x}\|_2^2 \quad (i = 1, \dots, n) \quad (37)$$

と定義する。式 (37) は元の点群における各点の再構成点群に対する最近傍距離を表している。元の点群として異常データが入力された場合、正常データのみを学習したモデルにより再構成された点群は、正常データに近いものとなる。そのため、元の点群における各点から最近傍距離をとると、元の点群において異常な点（座標として外れた点）での値は大きくなる。したがって、この値を各点の異常度として考えることができ、値に応じて元の点群の各点に色を付すことで異常度マップが作成できる。実際の運用時には、実務的な見地から、どの程度の異常を対象とするかにより適切な閾値を定め、それを上回ったものを異常と判定する。ここまでの説明を図-9 に示しているので、併せて参照されたい。

## 5. おわりに

本研究では、斜面・法面を対象とする点群データを用いて、それらの局所的異常を検知する点群深層学習モデルを開発した。具体的には、異常検知に用いられる深層学習モデルである AutoEncoder に対して、点群データを直接入力として取り込めるように拡張し、各点における異常度を算出する異常検知モデルを提案した。一方で、今後の課題として、オクルージョン（障害物）への対応があげられる。現場における法面によっては、植生等によりレーザーが遮蔽され、点群データの一部が欠損してしまうことがあるため、適切な方法で点群データを補完する必要がある。また、大域的異常を検知するモデルの開発も課題としてあげられる。本研究では、法面における局所的異常のみを対象として異常検知モデルを構築したが、実際の斜面では斜面全

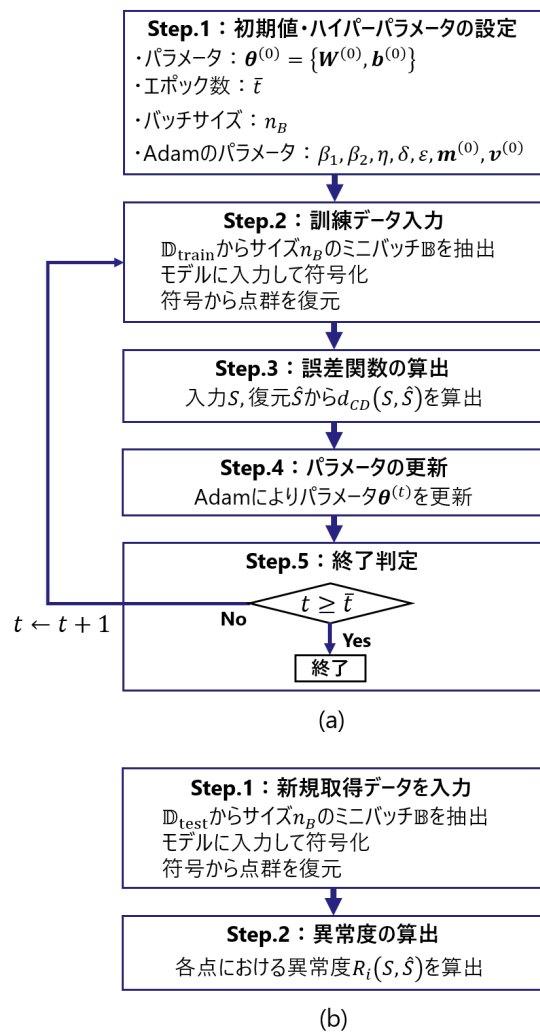


図-9 計算フローの概要。

(a) 訓練フロー、(b) 異常検知フロー。

体として変動がおこるような異常も存在し、方法論を修正していく必要があるといえる。

なお、本研究で提案した手法に基づいて、実際の法面点群データにおける実証分析を行っており、その結果及び考察については研究発表会にて提示する。

## 参考文献

- 1) 小林潔司：分権的ライフサイクル費用評価と集計的効率性，土木学会論文集，No.793/IV-68，pp.59-71，2005.
- 2) 貝戸清之，保田敬一，小林潔司，大和田慶：平均費用法に基づいた橋梁部材の最適補修戦略，土木学会論文集，No.801/I-73，pp.83-96，2005.
- 3) 貝戸清之，金治英貞，小林寛，間嶋信博，大石秀雄，松岡弘大：目視点検データを用いたフォルト・ツリー分析に基づく長大橋の最適点検政策の決定手法，土木学会論文集 F4，Vol.67，No.2，pp.74-91，2011.
- 4) 貝戸清之，金治英貞，杉岡弘一，大石秀雄，松岡弘大：鋼材腐食ハイブリッド劣化予測とフォルト・ツリー分析を用いた長大橋の最適点検間隔評価，土木学会論文集 F4，Vol.69，No.2，pp.84-101，2013.
- 5) 津田尚嵐，貝戸清之，青木一也，小林潔司：橋梁劣化予測のためのマルコフ推移確率の推定，土木学会論文集，

- No.801/I-73, pp.69-82, 2005.
- 6) 小濱健吾, 岡田貢一, 貝戸清之, 小林潔司:劣化ハザード率評価とベンチマーキング, 土木学会論文集 A, Vol.64, No.4, pp.857-874, 2008.
  - 7) 堤浩志, 小濱健吾, 中村葵, 小泉圭吾:斜面管理における RTK-GNSS データの変化点検知手法, 土木学会論文集 F4 (建設マネジメント) 特集号, Vol.75, No.2, L15-L26, 2019.
  - 8) 小林潔司, 貝戸清之, 水谷大二郎, 坂井康人:長期記憶性を考慮した時系列データによる構造物劣化過程のモデル化, 土木学会論文集 D3, Vol.75, No.4, pp.233-249, 2019.
  - 9) Qi, C. R., Su, H., Mo, K. and Guibas, L. J. : PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *Computer Vision and Pattern Recognition*, pp.652-660, 2017.
  - 10) Qi, C. R., Yi, L., Su, H. and Guibas, L. J. : PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, *Advances in Neural Information Processing Systems*, pp.5105-5114, 2017.
  - 11) Xu, Y., Fan, T., Xu, M., Zeng, L. and Qiao, Y. : SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters, *Proceedings of the European Conference on Computer Vision(ECCV)*, 2018.
  - 12) Yang, Y., Feng, C., Shen, Y. and Tian, D. : FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018.
  - 13) Dai, H. and Shao, L. : PointAE: Auto-encoder for 3D Statistical Shape and Texture Modelling, *International Conference on Computer Vision*, pp.5410-5419, 2019.
  - 14) Zhang, W., Jiang, H., Yang, Z., Yamakawa, S., Shimada, K. and Kara, L. B. :Data-driven Upsampling of Point Clouds, *arXiv preprint arXiv:1807.02740*, 2018.
  - 15) Fan, H., Su, H. and Guibas, L. : A Point Set Generation Network for 3D Object Reconstruction from a Single Image, *arXiv preprint arXiv:1612.00603*, 2016.
  - 16) 浜田成一, 貝戸清之, 杉原栄作, 水谷大二郎:元請負業者の施工要素技術の空洞化と経時的進行状況, 土木学会論文集 F4, Vol.77, No.1, pp.135-152, 2021.7
  - 17) 貝戸清之, 小林潔司:ビッグデータによるインフラマネジメント:アセットメトリクスにむけて, 土木学会論文集 D3 (土木計画学), 土木学会, Vol.70, No.5, pp.L21-L30, 2014.
  - 18) AI 白書編集委員会: AI 白書 2020, KADOKAWA, 2020.
  - 19) 全邦釘, 嶋本ゆり, 大窪和明, 三輪知寛, 大賀水田生:ディープラーニングおよび Random Forest によるコンクリートのひび割れ自動検出手法, 土木学会論文集 F3, Vol.73, No.2, pp.L297-L307, 2017.
  - 20) 橋本申, 吉井稔雄, 坪田隆宏, 全邦釘:ディープラーニングを用いた高速道路の路面劣化予測モデル, 土木学会論文集 D3, Vol.75, No.6, pp.L547-L554, 2020.
  - 21) 国土交通省:令和 2 年度に取り組む技術テーマについて, 2020.
  - 22) 国土交通省道路局:道路土工構造物点検要領, 2017.
  - 23) 井手剛, 杉山将:異常検知と変化検知, 講談社, 2015.
  - 24) Cohen, W. W.: Fast Effective Rule Induction, *Proceedings of the 12th International Conference on Machine Learning(ICML)*, pp.115-123, 1995.
  - 25) 立花亮介, 松原崇, 上原邦昭:深層生成モデルによる非正規化異常度を用いた工業製品の異常検知, 人工知能学会全国大会論文集, Vol.32, No.2A1-03, 2018.
  - 26) Bergmann, P., Lowe, S., Fauser, M., Sattlegger, D. and Steger, C.: Improving Unsupervised Defect Segmentation by Applying Structural Similarity To Autoencoders, *14th International Conference on Computer Vision Theory and Applications*, pp.372-380, 2019.
  - 27) Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. : Generative adversarial nets, *In Advances in Neural Information Processing Systems*, pp.2672-2680, 2014.
  - 28) Schlegl, T., Seebock, P., Waldstein, S. M., Schmidt-Erfurth, U. and Langs, G. : Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery, *arXiv preprint arXiv:1703.05921*, 2017.
  - 29) Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D. and Chen, H. : Deep autoencoding gaussian mixture model for unsupervised anomaly detection, *International Conference on Learning Representations*, 2018.
  - 30) Ruff, L., Vandermeulen, R. A., Gornitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Muller, E. and Kloft, M. : Deep One-Class Classification, *International Conference on Machine Learning*, pp.4393-4402, 2018.
  - 31) Glorot, X. and Bengio, Y. : Understanding the difficulty of training deep feedforward neural networks, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp.249-256, 2010.
  - 32) He, K., Zhang, X., Ren, S. and Sun, J. :Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, *Proceedings of the IEEE international conference on computer vision*, pp.1026-1034, 2015.
  - 33) Ioffe, S. and Szegedy, C. :Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *International Conference on Machine Learning*, pp.448-456, 2015.
  - 34) LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. : Gradient-Based Learning Applied to Document Recognition, *Proc. IEEE*, 1998.
  - 35) Goodfellow, I., Bengio, Y. and Courville, A. (岩澤有祐, 鈴木雅大, 中山浩太郎, 松尾豊監訳):深層学習, KADOKAWA, 2018.
  - 36) 岡谷貴之:深層学習, 講談社, 2015.
  - 37) Hinton, G. E. and Salakhutdinov, R. R. : Reducing the Dimensionality of Data with Neural Networks, *Science*, pp.504-507, 2006.
  - 38) 麻生英樹:多層ニューラルネットワークによる深層表現の学習, 人工知能学会誌, Vol. 28, No. 4, pp.649-659, 2013.
  - 39) Maturana, D. and Scherer, S. : VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.
  - 40) Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J. : 3D ShapeNets: A Deep Representation for Volumetric Shapes, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1912-1920, 2015.
  - 41) Kingma, D. P. and Ba, J. : Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*, 2014.