修士論文

ニューラルネットワークのパラメータの変化に着目した誤った教師データの検出手法

三島 惇也

2024 年 12 月 3 日

岐阜大学大学院 自然科学技術研究科 知能理工学専攻 知能情報学領域
鈴木研究室

本論文は岐阜大学大学院 自然科学技術研究科に
修士（工学）授与の要件として提出した修士論文である.

三島 惇也

指導教員：
鈴木 優 准教授

ニューラルネットワークのパラメータの変化に着目した誤った教師データの検出手法*

三島 惇也

内容梗概

本研究ではデータクレンジングの精度向上を目的とした，外れ値検出手法を提案する．我々は n クラス分類を行うことが可能な学習済みのニューラルネットワークについて，あるクラス a の分類境界内に存在する教師データのうち，教師ラベルがクラス a 以外のものは，クラス a の分類境界から遠いほど誤りである可能性が高いという仮説を立てた．仮説を基に手法を考案する際，分類器の分類境界を知るためには膨大な時間が掛かるという問題がある．分類境界を直接知る方法がないため，我々は入力される可能性のあるデータを全パターン用意するしかない．そこで，我々は学習時に更新される分類器のパラメータの変化に着目し，分類境界からの距離と相関があると考えられる特徴を現実的な時間で得ることを考えた．提案手法が外れ値検出の精度を AUROC によって評価した．その結果，比較手法と比較した条件全 40 条件のうち 28 条件（70%）で提案手法が勝利した．

キーワード

データクレンジング，外れ値検出，ニューラルネットワーク，機械学習，特徴量抽出

*岐阜大学大学院 自然科学技術研究科 知能理工学専攻 知能情報学領域 修士論文, 学籍番号: 1224525091, 2024 年 12 月 3 日.

目次

図目次

表目次

第 1 章　はじめに

教師あり学習 [1] において，機械学習モデルを高精度に構築するためには，正しい教師データ（入力データと教師ラベルの組）を持つことが重要である．しかし公開されているデータセットの中には，入力データと教師ラベルの組が正しくない教師データ（以下，誤った教師データと呼称）が存在する [2]．データセット内に含まれる教師データの数が多いとき，人手ですべてのデータを精査するためには多くの時間と労力がかかる．そこで我々は労力を軽減するために，データクレンジングによって，誤った教師データを削除したデータセットを作成する必要があると考えた．我々は既存のデータクレンジング手法の精度向上を目的とした研究を行う．本稿では画像分類だけを扱う．

既存のデータクレンジング手法には外れ値検出手法を用いたものがある．しかし，本稿で扱う画像のデータセットではベクトルの次元数が膨大となってしまうため，以下に示す二つのデメリットが増す．一つ目は次元の呪いによって外れ値検出の感度が弱くなることである．次元の呪い（Curse of dimension）は，あるデータの次元数が増えれば増えるほど，データが空間の外側に分布する現象のことである．二つ目は外れ値検出の処理にかかる時間が長くなることである．これらの欠点を解決するために，本稿では，特徴量抽出を用いて次元削減を行い，その特徴量を外れ値検出に使用する方法を提案する．

我々は，複数のクラス $a, b, \cdots$ へデータを分類する問題を考えたとき，あるクラス a の分類境界外に存在する教師データのうち教師ラベルがクラス a である教師データは，クラス a の分類境界から遠いほど誤った教師データである可能性が高いという仮説を立てた．ここで分類境界とは，与えられたデータセットを用いて構築した機械学習モデル（以下，学習済みモデルと呼称）の分類境界のことを指す．本稿では，クラス a とそれ以外のクラスを分ける分類境界が存在するとき，クラス a であると判定される側の領域を分類境界内，それ以外のクラスであると判定される側の領域を分類境界外と呼ぶ．基本的な考え方を図 1.1 を用いて説明する．図 1.1 は，白から黒へのグラデーション（下段）を部分的に見たときに白か黒かを分類するタスクを解いた例である．真の正解はちょうど中間の部分で白と黒に分かれているのに対して，教師ラベルは 白$_1$，白$_2$ と書かれた箇所も白色になっている．この

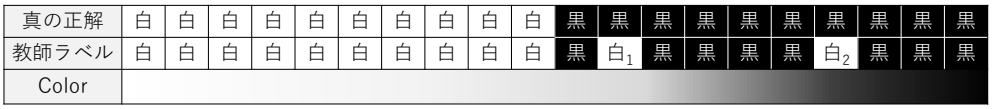

図 1.1　白から黒へのグラデーションに対する白黒判別を行った際の真の正解と教師ラベル

とき，真の正解の白と黒の境界が分類境界に相当し，白$_1$，白$_2$ と書かれた箇所はそれぞれ，クラス a の分類境界に近い教師データと遠い教師データに相当する．白$_1$ と書かれた箇所の Color を見ると，白に分類される色とあまり変わらないように見えるため，誤りと断言できない．しかし，白$_2$ と書かれた箇所の Color を見ると，黒に分類される色に近いように見えるため，誤りと断言しても良い．我々は，上記で説明したような状態が学習済みモデルの分類境界にも存在するのではないかと考えた．

そこで我々は，クラス a の分類境界外に存在する教師データのうち，教師ラベルがクラス a である教師データと分類境界の距離を測定することを考えた．しかし，機械学習モデルの分類境界を直接測定方法は存在しないため，教師データと分類境界の距離を測ることができない．グリッドサーチのような探索的な方法で分類境界を推定することは可能だが，入力される可能性のあるデータをすべて用意する必要があり，処理時間が膨大になるため現実的ではない．そのため，我々は分類境界の代替となる特徴量が必要であると考えた．

我々はニューラルネットワークの分類境界の代替となる特徴量はニューラルネットワークのパラメータであると考えた．分類境界が更新される時にはニューラルネットワークのパラメータ（重みやバイアス）が変化する．そのため，ニューラルネットワークのパラメータは分類境界を表現していると考えた．上記の考えを基に，我々は以下の手順によって，ニューラルネットワークの分類境界に相当する特徴量を抽出する．

1. クラス a の分類境界外に存在する教師データのうち，教師ラベルがクラス a である教師データを一つ用意する．
2. 用意した教師データを用いて学習済みモデルのパラメータを更新する．
3. 更新後の分類境界の変化をパラメータから取得する．

得られた特徴量と元のニューラルネットワークのパラメータを比較することによって，分類境界がどの程度変化したかを知ることができる．そして，我々はこの特徴量を用いて外れ値検出を行うことによって，誤った教師データを検出することができると考えた．

提案手法が誤った教師データを検出する手法として優れていることを示すために，Confident Learning[3]，Label Fix[4] との比較実験を行った．その結果，提案手法は最大で AUROC が約 9% 向上しており，比較手法よりも精度が高いことを確認した．

また, 提案手法の適用範囲を調べる過程で, 我々はどんなニューラルネットワークモデルであっても提案手法が適用できることを確認した．しかし，ニューラルネットワークモデルの種類によっては性能が低下するという提案手法の問題点を発見した．この問題を解決するために我々は複数のニューラルネットワークモデルを用意し，性能比較実験を行った．その結果，ニューラルネットワークモデルの種類によって性能が低下する原因がニューラルネットワークの Batch Normalization であることを特定した．ミニバッチに含まれる教師データの分布は，学習済みモデル構築時のミニバッチと，特徴量抽出時のミニバッチで異なる．Batch Normalization があることによって性能が低下する原因は，この違い原因であると考察した．そして，我々は上記の考察を基に提案手法の問題点に対応するために改善を行った．改善後の特徴量の算出方法を以下に示す．

1. クラス a の分類境界外に存在する教師データのうち，教師ラベルがクラス a である教師データを一つ用意する．
2. 用意した教師データを学習済みモデル構築時に使用した訓練データに混ぜたミニバッチを使用して学習済みモデルのパラメータを更新する．
3. 更新後の分類境界の変化をパラメータから取得する．

我々は手順の 2. に変更を加えた．学習済みモデル構築時に使用した訓練データのミニバッチを使用し，用意した教師データをそのミニバッチに混ぜるように変更した．変更前はミニバッチに含まれる教師データが全て一つの教師データによって構成されていたが，変更後はミニバッチに含まれる教師データが複数の教師データによって構成されるようになった．我々はこの変更を行うことによって，学習済みモ

デル構築時のミニバッチと，特徴量抽出時のミニバッチは教師データの分布が異なるという問題を軽減することが可能であると考えた．その結果我々は，問題が解消され，多くの種類のニューラルネットワークにおいて性能 (AUROC) が低下すること無く提案手法が利用可能であることを確認した．

本研究の貢献は以下のとおりである．

- 誤った教師データの検出に有用な特徴量抽出手法を提案した．
- 比較手法よりも精度の高い誤った教師データの検出手法を提案した．

第 2 章　基本的事項

本章では，本稿で用いた技術，手法の説明を行う．

2.1　機械学習

機械学習とは，機械学習モデルに大量のデータを入力することによって，機械学習モデルが自動的に学習する方法である．機械学習を行うことができる仕組みを機械学習モデルと呼び，機械学習モデルに入力する大量のデータをデータセットと呼ぶ．機械学習モデルはデータセットから得られる情報から，パターンや分類するためのルールなど，問題の解き方を学習する．機械学習には大きく分けて三つの学習方法がある．一つ目は教師あり学習，二つ目は教師なし学習，そして三つ目は強化学習である．

教師あり学習は機械学習モデルにデータを入力する際，機械にそのデータを入力したときに期待する出力を同時に教える学習方法である．つまり，教師あり学習は，入力されたデータと正解を確認しながら学習を行う方法である．例えば，教師あり学習のための機械学習モデルには線形回帰，重回帰，ロジスティック回帰，ニューラルネットワークといった機械学習モデルがある．このとき，入力するデータのことを入力データ，教師あり学習のために教える期待する出力のことを教師ラベルと呼ぶ．また，入力データと教師ラベルの組を教師データと呼ぶ．

教師なし学習は機械学習モデルに入力された大量のデータのみを用いて，データの分類，情報の抽出方法を自動的に学習させる学習方法である．つまり，教師なし学習は，入力データのみを使用して学習を行う方法である．例えば，k-Means 法や階層型クラスタリングといったクラスタリング手法，t-SNE や PCA といった次元削減手法は教師なし学習に属する機械学習モデルである．

強化学習は上記の二つの学習方法とは大きく異なる．強化学習では，行動を行うエージェントという存在を作り，学習を行う．機械学習モデルに入力されたデータから，エージェントが行動を選択する．機械学習モデルが行った行動を評価し，報酬を与える．機械学習モデルは得られる報酬が最大化するように学習を行う．以上に示した方法で強化学習用の機械学習モデルを実現している．つまり，強化学習は，

機械に行動を選択させてから，フィードバックを行うことによって学習を行う方法である．強化学習において，与える報酬を決定する関数を報酬関数と呼ぶ．

本稿で提案するデータクレンジング手法は，まずニューラルネットワークモデルを使用した教師あり学習を行い，特徴量抽出をした後，教師なし学習による外れ値検出を行う．提案手法の詳細は 4 章や 5 章で述べる．

2.2 機械学習のタスク

機械学習において，機械学習モデルが解く問題をタスクと呼ぶ．このタスクには，大きく分けて 3 種類のタスクが存在する．一つ目は分類タスク，二つ目は回帰タスク，三つ目は推薦タスクである．

分類タスクは入力データがどのカテゴリ，離散値に属するデータかを判別するタスクである．そのため，分類タスクを解く機械学習モデルは入力データが属するカテゴリを出力する．本稿で使用するデータセットは全て画像の分類タスクを学習するために用意されたデータセットである．

回帰タスクは入力データからある値を予測するタスクである．そのため，回帰タスクを解く機械学習モデルは，予測した実数値を出力する．例えば，家の価格を予測するデータセットが公開されている．

推薦タスクはユーザの好みを学習し，ユーザ好みに一致するものを提示するタスクである．そのため，推薦タスクを解く機械学習モデルはユーザの好みに一致する上位何件かを出力する．

2.3 損失関数

損失関数とは，機械学習モデルの出力と教師ラベルの誤差を表す関数である．損失関数には平均二乗誤差 (MSELoss，Mean Square Error Loss)，Cross Entropy Loss などいろいろな損失関数が考案されている．損失関数は値が小さいほど誤差が少なく，機械学習モデルの予測が教師データに適合していることが確認できる．

平均二乗誤差は回帰タスクに使用される損失関数である．平均二乗誤差の式は，データセットの教師データ数を N，i 番目の教師データに対する機械学習モデルの

予測を $\hat{y}^{(i)}$，教師ラベルを $y^{(i)}$ とすると，

$$MSELoss = \frac{\sum_{i=1}^{N}(y^{(i)} - \hat{y}^{(i)})^2}{N} \tag{2.3.1}$$

で表される．

Cross Entropy Loss は分類タスクに使用される損失関数である．Cross Entropy Loss の式は，k クラス分類を行うためのデータセット（教師データ数は N）を用いる場合，i 番目の教師データのクラス j に対する機械学習モデルの予測を $\hat{y}_j^{(i)}$，i 番目の教師データのクラス j に対する教師ラベルを $y_j^{(i)}$ とすると，

$$CrossEntropyLoss = \sum_{i=1}^{N}\sum_{j=1}^{k} y_j^{(i)} log(\hat{y}_j^{(i)}) \tag{2.3.2}$$

で表される．

これらの損失関数は微分可能であり，損失関数の値を小さくするために最小二乗法，最急降下法など，最適化手法を用いてパラメータ調整を行う．$\hat{y}$ で表されている変数はそれぞれ機械学習モデルのパラメータと入力データから算出された値である．そのため，$\hat{y} = f$(入力データ) と考えることができる．f(入力データ) を微分し，損失関数が小さくなるように f(入力データ) 内のパラメータを更新することによって，機械学習モデルの学習が進む．

2.4 データクレンジング

機械学習に限らず，あるデータが与えられたときに本来あるはずのデータが存在していなかったり (欠損値)，明らかに誤りであるデータがしていたり (誤った情報の混入) することがある．上記のような欠損値，誤った情報を削除したり，修正したりすることをデータクレンジングという．

1 章で述べた通り，我々は誤った教師データを削除したデータセットを作成する必要があると考えている．本稿において，我々は誤った教師データが含まれるデータセットは誤った情報の混入が起きていると考える．よって，本項におけるデータクレンジングの定義は，“誤った教師データを特定し，削除する，修正するなどの問題解決を行うこと” とする．本研究と同様の問題設定で行われているデータクレンジング手法の詳細については，3 章で述べる．

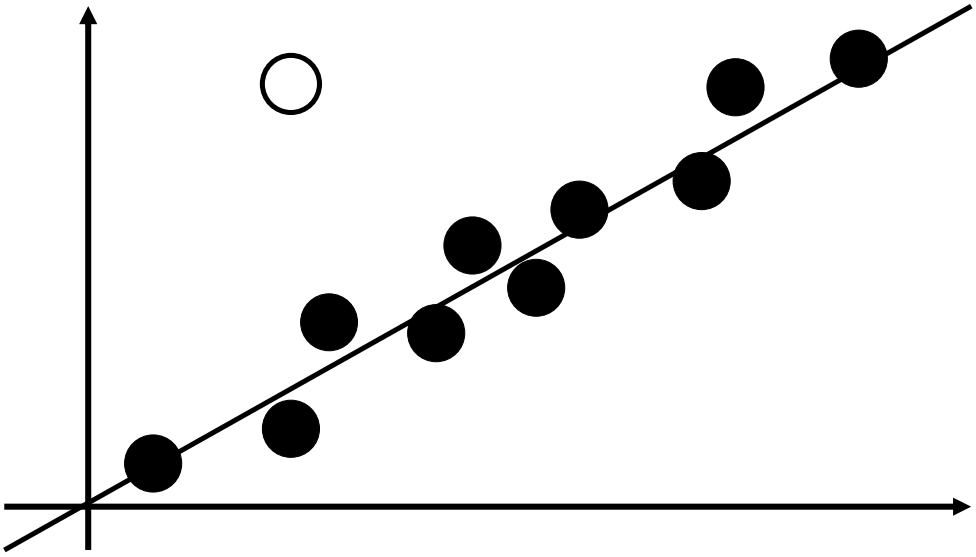

図 2.1　回帰タスクにおける外れ値の例

2.5　外れ値検出

外れ値検出とは，外れ値を見つけるための仕組みである．外れ値は文献 [5] によると，“外れ値とは，データの大部分の傾向と異なるもので，必ずしも誤りとは限らないが，データ集計や分析の際にその存在が結果をゆがめる可能性がある” と定義されている．この定義を基に，機械学習における外れ値を回帰タスクと分類タスクの例を用いて説明する．

回帰タスクにおける外れ値は図 2.1 に示した白丸のような値である．図 2.1 は直線を用いて回帰タスクを解いた時の回帰結果（直線）と教師データ分布（丸）を表している．黒丸で表されている教師データは直線に近い箇所に分布しているが，白丸は黒丸と比較して直線から離れている．この白丸の教師データは外れ値の定義に当てはまると考えられるため，白丸のような教師データを外れ値と呼ぶ．

分類タスクにおける外れ値は図 2.2 に示した白丸のような値である．図 2.2 は分類タスクを解いた時の分類境界（大きい丸）と教師データ分布（黒塗りの図形と白丸一つ）を表している．実線は黒丸のクラス，点線は三角のクラス，二重の実線はばつ印のクラス，二重の点線は四角のクラスの分類境界を表している．しかし，白

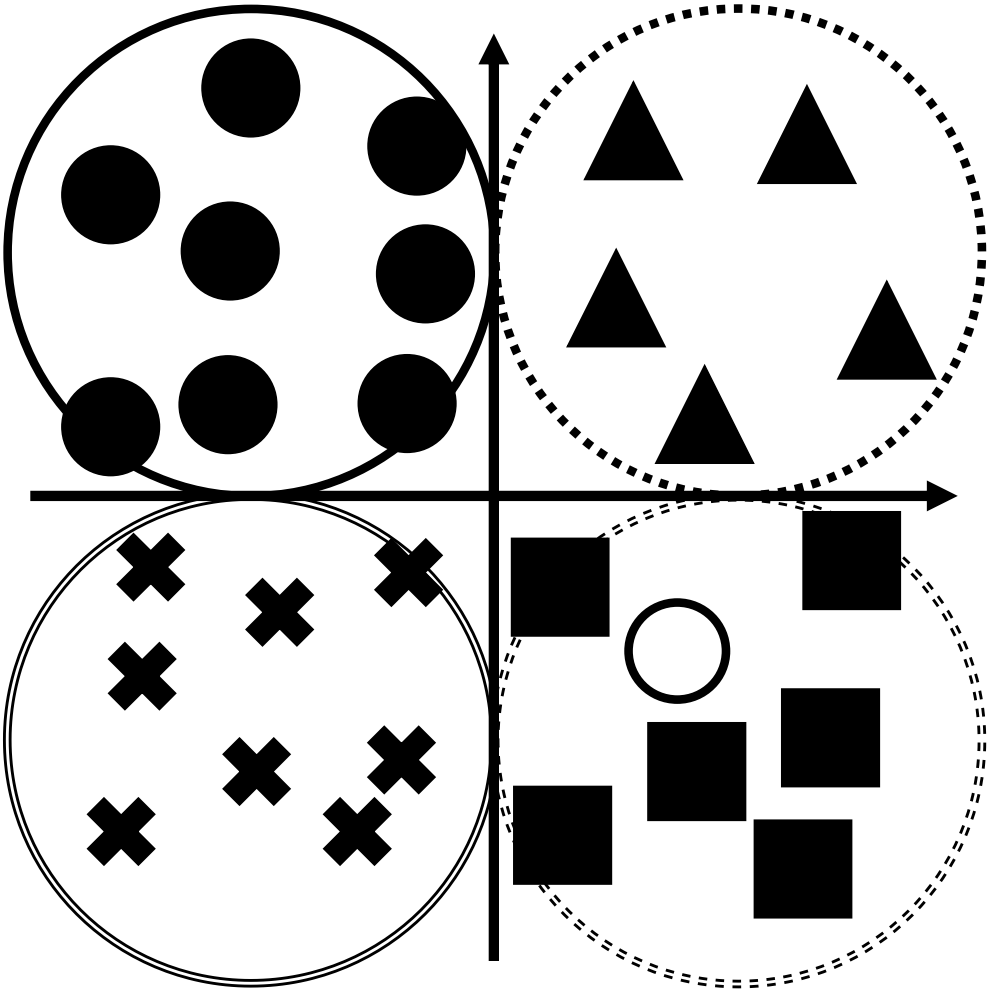

図 2.2　分類タスクにおける外れ値の例

丸は黒丸の分類境界から離れており，四角の分類境界内に入り込んでしまっている．この白丸の教師データは外れ値の定義に当てはまると考えられるため，白丸のような教師データを外れ値と呼ぶ．

上記の回帰結果と分類境界は我々が説明のために考えたものである．機械学習を行った際に上記と同様の結果になるとは限らないことに注意されたい．本稿で用いるデータセットは分類タスク用データセットであるため，我々は上記の外れ値のうち後者を検出する手法を提案する．

2.6 次元の呪い

機械学習において，次元の呪いといわれるものは二つ存在している．これらの次元の呪いは，Curse of dimensionality と Curse of dimension に呼び分けされている．1 章で述べた次元の呪いは後者である．

まず，Curse of dimensionality について説明する．Curse of dimensionality は，教師あり学習において，機械学習モデルのパラメータを増やすと，学習に必要な教師データが指数関数的に増加することを指す．機械学習において，性能向上のためには，パラメータを増やす（入力データの次元数を増やすことを含む），もしくは教師データを増やすことが必要である．パラメータを増やすと過学習を起こすリスクが増加する．過学習とは，教師データに対して機械学習モデルが過剰に適合し，汎化性能のない（未知データを正解できない）モデルが出来上がる現象である．そして，この過学習という現象を防ぐためには教師データを増やす必要がある．パラメータを増やせば増やすほど，過学習を防ぐために必要な教師データは多くなる．そのため，パラメータの過不足と教師データの過不足はトレードオフの関係にあるといえる．この関係を表す言葉が，Curse of dimentionality である．

次に，1 章で述べた次元の呪いを指す Curse of dimension について説明する．Curse of dimension は，あるデータの次元数が増えれば増えるほど，データが空間の外側に分布する現象を指す．3 次元以上の球体には以下の二つの特徴がある．

1. 次元数が増えれば増えるほど，超球の表面付近の体積と全体の体積に差がなくなっていく．
2. 次元数が増えれば増えるほど，任意の点からの最近傍点との距離と，最遠傍点との距離の差が 0 に近づく．

そのため，次元数が増えれば増えるほどデータ間の距離に差がないように見えてしまうという．本稿で扱う外れ値検出手法は，分布の中心点 (平均) からどれだけ距離が離れているかを用いて評価する手法である．そのため，画像などの高次元データをそのまま利用すると，距離に差がないように見えてしまい，外れ値の検出がうまくいかないという問題がある．そのため我々は，1 章で述べたように次元数を減らす方法を適用する必要があると考えた．

3 次元以上の球体に存在する二つの特徴について，簡単に説明する．n 次元の球

体の体積 ($V(n)$) を求める式は，半径を r とすると，

$$V(n,r) = \frac{n\pi^{\frac{n}{2}} r^n}{\Gamma(\frac{n}{2}+1)} \tag{2.6.1}$$

で表すことができる．単位球 ($r=1$) と，少し内側の球（例えば，$r=0.9$）の体積の差を考える．この差は

$$V(n,1) - V(n,0.9) = \frac{n\pi^{\frac{n}{2}} 1^n}{\Gamma(\frac{n}{2}+1)} - \frac{n\pi^{\frac{n}{2}} 0.9^n}{\Gamma(\frac{n}{2}+1)} \tag{2.6.2}$$

と表され，共通因数をまとめると

$$V(n,1) - V(n,0.9) = \frac{n\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}(1^n - 0.9^n) \tag{2.6.3}$$

となる．このとき，n を無限に設定すると，$1^\infty = 1$，$0.9^\infty = 0$ であるため，

$$V(\infty,1) - V(\infty,0.9) = \frac{\infty\pi^{\frac{\infty}{2}}}{\Gamma(\frac{\infty}{2}+1)}(= V(\infty,1)) \tag{2.6.4}$$

となる．つまり，半径が 1 の球は，次元数の増加と共に，その内側に存在する半径 0.9 の球が持つ体積が減少し，外側に体積が集中していくといえる．そのため，3 次元以上の球体には上記の二つの特徴があるといえる．

表 2.1　データセットの例（住宅価格予測用のデータセット）

	住民所得 (万)	築年数	部屋数	寝室数	人口	世帯人数	住宅価格 (万)
Block1	1,000	40	7	1	400	3	4,000
Block2	850	20	6	1	1,200	4	3,500
Block3	900	30	8	1	600	3	3,000
Block4	600	10	5	1	2,400	5	2,500
Block5	400	40	4	1	800	4	2,000
Block6	350	50	5	1	650	4	1,200

2.7　次元削減

次元削減とは，入力データから得られる情報量を維持しつつ，入力データの次元数を減らす方法である．次元削減は大きく分けて二つの方法がある．一つ目は特徴量選択，二つ目は特徴量抽出である．

例えば，表 2.1 に示したデータセットが与えられたとする．表 2.1 は住宅価格を予測する回帰タスク用データセットを想定した例である．Block の番号ごとに地域が異なり，各項目の平均値が与えられているとする．住宅価格を予測するための単純な方法は，住宅価格以外の項目のデータを全て使用して予測を行うことである．このとき，教師データの数が少ないために，過学習が起きたとする．その場合，データ数は増やせないため，入力データの次元数を減らす必要がある．この例を用いて特徴量選択，特徴量抽出を用いた次元削減方法を説明する．

特徴量選択は，回帰に使用する入力データの項目を減らす方法である．例えば，表 2.1 を見ると，寝室数は全て 1 となっており，入力する必要がないと考えられる．機械学習モデルに入力するデータに寝室数を含まないようにすることによって，入力データの次元数を 1 削減することができる．このような方法を取る次元削減方法を特徴量選択という．

特徴量抽出は，回帰に使用する入力データの情報量を落とさないようにしつつ次元を減らす方法である．例えば，表 2.1 を見ると，Block の人口と世帯人数という項目がある．これらの項目をまとめるために 人口 ÷ 世帯人数 を計算すると世帯数という項目になる．人口と世帯人数に分かれた状態と比べると世帯数は情報量は落ちているが，両方の情報を含む特徴量に変換されている．このような変換を行う方法が特徴量抽出である．本研究はニューラルネットワークを用いて特徴量抽出を行う手法である．

2.8　マハラノビス距離

マハラノビス距離とは，平均，共分散行列を用いて，データの分布に基づいた等高線を引くようなイメージ（図 2.3）の距離関数である．マハラノビス距離は，平均と分散共分散行列を用いてデータの分布に基づいた距離を算出する．この距離はデータの分布の中心（平均）から楕円形に等高線が広がっていく距離である．イ

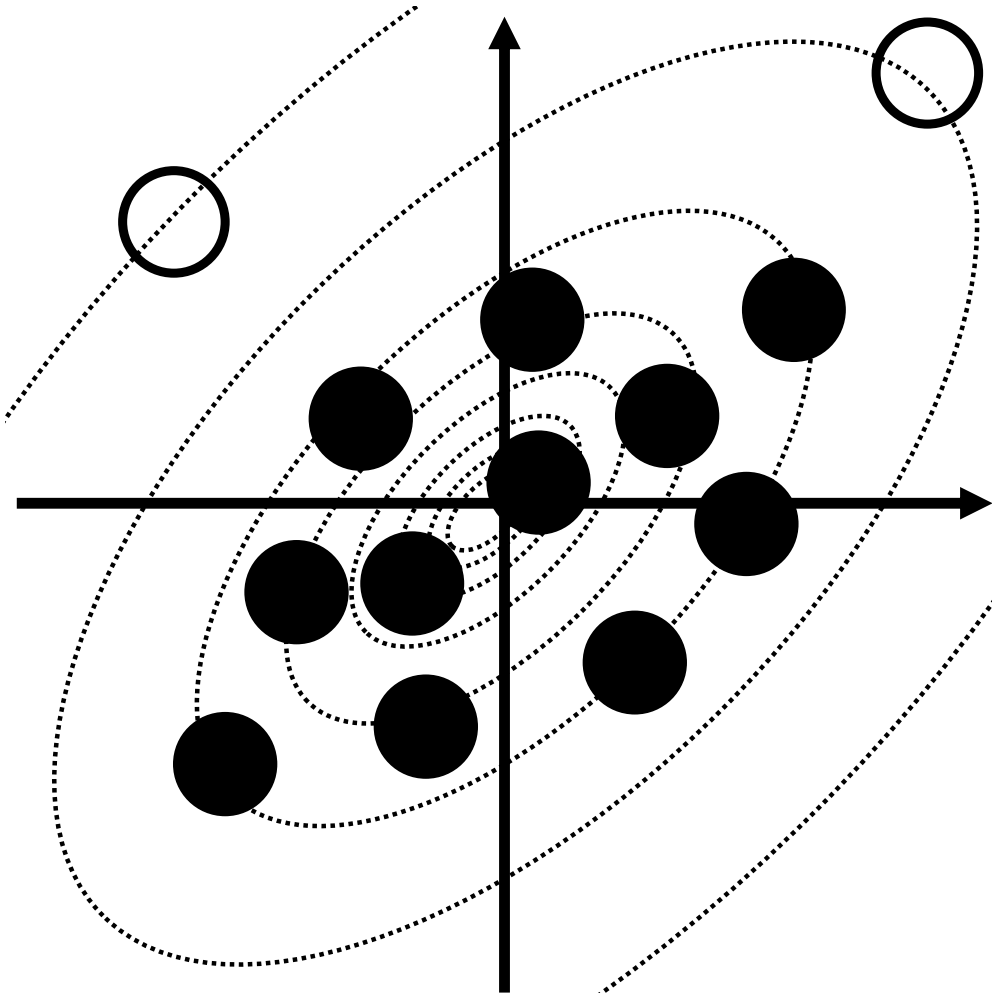

図 2.3　マハラノビス距離のイメージと外れ値 (白丸) の例

メージ図は図 2.3 に示した通りである．マハラノビス距離を用いた外れ値検出では，一定距離以上離れたデータを外れ値として判定する．

マハラノビス距離の計算式は,

$$\sqrt{(x-\mu)^T\Sigma^{-1}(x-\mu)} \tag{2.8.1}$$

で，この時の x が任意のデータの座標,μ がデータ全体の平均座標，Σ が共分散行列である．データの分布が楕円形である場合も外れ値の距離が大きくなるという特徴がある．図 2.3 に示した分布の場合は白丸のような値が外れ値と判断される．どの

程度距離が離れていたら外れ値と判断するかは，任意の値に設定しなければならない．その設定を閾値と呼ぶ.

2.9 評価指標

本稿にて使用する評価指標について，True と False の 2 値分類を行う場合を想定した例を示しながら説明する．2 値分類を行う場合，表 2.2 に示したような予測の結果と正解の組ができる.

予測の結果が True であり，正解も True であるものを True Positive（真陽性，TP）といい，正解は False であるものを False Positive（偽陽性，FP）という．また，予測の結果が False であり，正解は True であるものを False Negative（偽陰性，FN）といい，正解が False であるものを True Negative（真陰性，TN）という.

2.9.1 Accuracy

Accuracy とは正解率のことである．例えば，10 個の入力データがあった場合，予測結果のうち 6 個が正しく，4 個が間違っていると Accuracy は 0.6 となる．表 2.2 の文字 (TP，FP，FN，TN) を用いて表すと以下に示す式になる.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.9.1}$$

表 2.2　予測の結果と正解の組み合わせ

	正解が True	正解が False
予測の結果が True	True Positive（真陽性,TP）	False Positive（偽陽性,FP）
予測の結果が False	False Negative（偽陰性,FN）	True Negative（真陰性 TN）

2.9.2 Precision

Precision は適合率とも呼ばれ，予測の結果が True だったときに，正解も True である割合である．表 2.2 を用いて式で表すと以下のとおりである．

$$Precision = \frac{TP}{TP + FP} \tag{2.9.2}$$

2.9.3 Recall

Recall は再現率，真陽性率といい，正解が True だったときに，予測の結果も True である割合である．表 2.2 を用いて式で表すと以下のとおりである．

$$Recall = \frac{TP}{TP + FN} \tag{2.9.3}$$

2.9.4 Fall-out

Fall-out は偽陽性率といい，正解が False だった時に，予測の結果も False である割合である．表 2.2 を用いて式で表すと以下のとおりである．

$$Fall\text{-}out = \frac{FP}{FP + TN} \tag{2.9.4}$$

2.9.5 AUROC

AUROC とは，ROC 曲線の曲線下面積 (AUC) のことを指す評価指標である．この評価指標は，ROC 曲線と曲線下面積 (AUC) の名称を結合して AUROC と呼ばれている．

ROC 曲線とは，予測結果を True または False と予測する閾値を変化させたときの Recall（縦軸）と Fall-out（横軸）の値をプロットした曲線のことである．すべて True と予測すると Recall，Fall-out がともに 1 となり，すべて False と予測すると Recall，Fall-out がともに 0 になる．閾値を変化させ，図 2.4 に示したような曲線を描くと，その曲線が ROC 曲線になる．ROC 曲線よりも下の部分の面積が

大きいほど良いモデルであるとされている．また，完全ランダムに予測を行った場合の AUROC は 0.5（$(0,0)$ から $(1,1)$ への直線）となる．

例として，真偽のラベルと予測スコアが表 2.3 であった場合について述べる．閾値を 0.1 ずつ変化させていったときの Recall と Fall-out の変化を表 2.4 に示す．表 2.4 に示した値をプロットし，その値を結び，図 2.4 のようなグラフを作成する．図 2.4 の横軸は偽陽性率 (Fall-out)，縦軸は真陽性率 (Recall)，実線で引かれた線が表 2.3 の ROC 曲線，破線で引かれた線がランダム予測時の ROC 曲線を表している．図 2.4 の ROC curve (area(%) = 93.75) の部分は，実線で引かれた ROC 曲線の AUC の値を示している．つまり，表 2.3 に示した Recall，Fall-out から算出される AUROC(%) の値は 93.75 である．

2.10　ニューラルネットワーク

ニューラルネットワークとは，人間の神経細胞であるニューロンとそのつながりを模してつくられた機械学習の手法である．人間のニューロン一つをパーセプトロンと呼ばれる人工ニューロンとして実装し，人間の神経回路と同様に複数個つなげてネットワークを構築するというものである．このようにして構築したネットワークをニューラルネットワークと呼ぶ．入力の特徴量が 4 個で 3 クラス分類のニュー

表 2.3　実際のラベルと予測の例

data number	True or False	score
1	True	0.9
2	False	0.1
3	True	0.8
4	False	0.6
5	False	0.2
6	True	0.4
7	False	0.3
8	True	0.7

ラルネットワークの例を図 2.6 に示す．このようなネットワークを構築し，学習を行うことによって，複雑な問題に対しても精度の高い予測，分類が可能となる．

ニューラルネットワークはパーセプトロンを多数結合したモデルであり，複雑な関数も近似することが可能であるとされている．ここで複雑な関数とは，回帰タスクであれば非線形な回帰曲線，分類タスクであれば非線形な分類境界のことを指している．ニューラルネットワークに対して入力データを入力する層を入力層（図 2.6 の左側），ニューラルネットワークが出力を出す層を出力層（図 2.6 の右側），それ以外の層を隠れ層（図 2.6 の中間）と呼ぶ．従来の機械学習の手法 (重回帰分析やロジスティック回帰など) よりも回帰タスクや分類タスクを解くための関数を複雑にできるため，従来の手法では解けなかった問題も解くことができる．

レコメンデーションや自動運転，文書分類などの分野において，ディープラーニングを用いた手法がよく利用されている．ディープラーニングとは，ニューラルネットワークのうち，隠れ層が多層構造になっているモデルで学習したニューラルネットワークモデルである．ディープラーニングを用いることで，隠れ層が 1 層の時よりもさらに複雑な問題にもうまく対応できるとされており，画像やテキス

表 2.4　表 2.3 から閾値ごとに求めた真陽性率と偽陽性率

閾値 (閾値未満を False と判断)	Recall	Fall-out
0.0(All True)	1.0	1.0
0.1	1.0	1.0
0.2	1.0	0.75
0.3	1.0	0.5
0.4	1.0	0.25
0.5	0.75	0.25
0.6	0.75	0.25
0.7	0.75	0.0
0.8	0.5	0.0
0.9	0.25	0.0
1.0(All False)	0.0	0.0

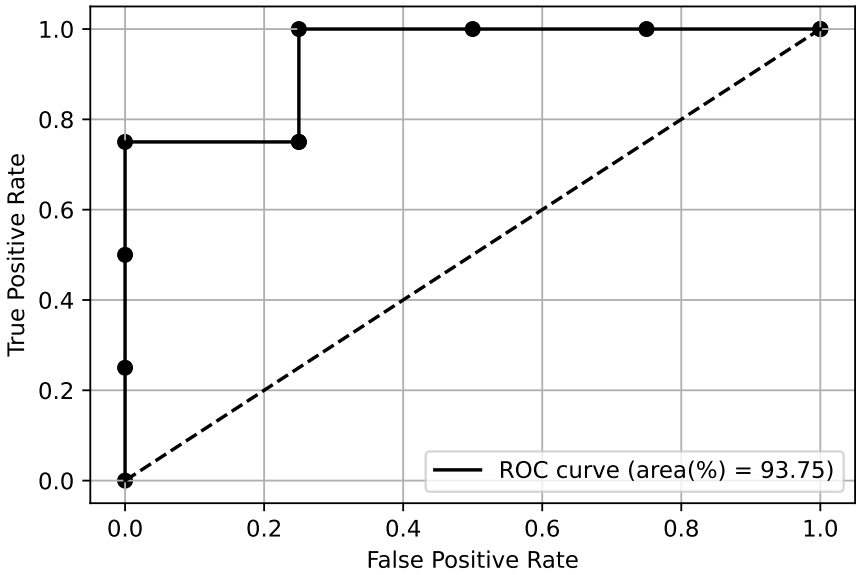

図 2.4　ROC 曲線の例

トなどの高次元データを扱う問題であっても利用できる．そのため，様々な分野においてディープラーニングを用いた手法が主流になりつつある．画像の分野では ResNet[6]，自然言語の分野では BERT[7] など，隠れ層を多層化して精度の高いモデルを作成している研究が数多く行われている．最近有名なディープラーニングを用いたサービスは ChatGPT が挙げられる．

ニューラルネットワークの学習について簡単に説明する．ニューラルネットワークは入力層，隠れ層 (中間層)，出力層に分かれており，各層の間にはニューロン同士の関係の強さを示す重みが存在する．人間の神経回路でいうシナプスの結合強度のことであり，重みはその結合強度を人工的に再現したものである．この重みによって，一つ前の層のパーセプトロンから入力された情報をどの程度重要視するかが決まる．図 2.5 で示した例は，入力として特徴量が 3 種類で True か False の 2 値分類のタスクを行う場合の例である．図 2.5 に示したように入力に対してそれぞれの入力の重要度を示す重み w_1, w_2, w_3 を与え，重みをかけて合計した値から出力

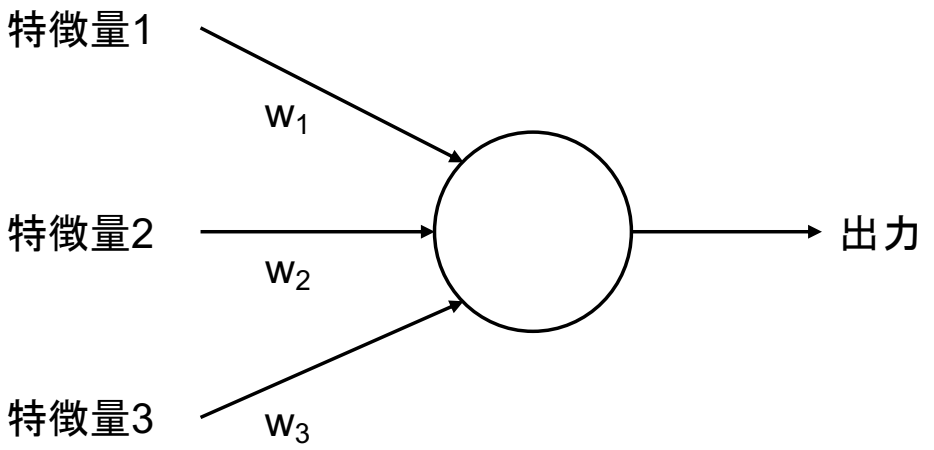

図 2.5　人工ニューロン（パーセプトロン）の例

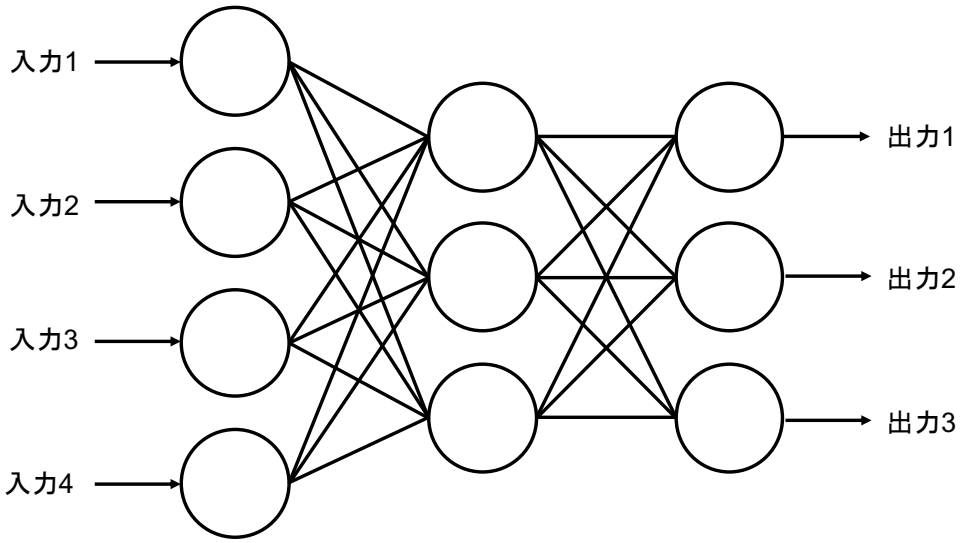

図 2.6　ニューラルネットワークの例

を得ている．このときの計算式は

$$\text{出力} = \text{特徴量 } 1 \times w_1 + \text{特徴量 } 2 \times w_2 + \text{特徴量 } 3 \times w_3 \tag{2.10.1}$$

である．また，バイアス項 b（切片）を追加する場合の計算式は

$$\text{出力} = \text{特徴量 } 1 \times w_1 + \text{特徴量 } 2 \times w_2 + \text{特徴量 } 3 \times w_3 + 1 \times b \tag{2.10.2}$$

である．この出力に活性化関数（ReLU 関数，SoftMax 関数，Sigmoid 関数など）をかけることによって，非線形な回帰，分類が可能となる．特に ReLU 関数の発見はニューラルネットワークの発展を加速させ，ディープラーニングが主流になっていくきっかけとなった．パーセプトロンの出力に活性化関数をかけたものを学習することはロジスティック回帰モデルを学習することと同じである．

ニューラルネットワークの学習はバックプロパゲーション（誤差逆伝搬）によって行われる．学習手順は大きく分けると以下の手順となる．

(1) 入力データをニューラルネットワークに入力し，予測を行う．
(2) 教師データと出力の誤差（損失関数）を算出する．
(3) 損失関数の値が小さくなるよう重みを更新する．

このときの (3) で重みを更新するための計算がバックプロパゲーションである．(2) では，教師データと実際の予測との誤差（2.3 節で説明した損失関数）を算出する．(3) では，損失関数が下がるように最急降下法などを用いて重みの更新を行う．この更新は予測とは逆の順番（出力層→隠れ層→入力層の順）で行われる．逆の順番になる理由は，教師データと予測結果から誤差を逆算する必要があるためである．ニューラルネットワークモデルは (1) から (3) の操作を繰り返すことで学習が進み，問題を解くことができるようになる．

2.11 k-分割交差検証

k-分割交差検証とは，教師あり学習の機械学習モデルの性能を測る際のデータセットの使用方法である．例えば，10,000 件の教師データからなるデータセットを用いて機械学習モデルの性能を測りたいとする．このとき，k-分割交差検証を行う場合，我々は以下に示すように機械学習の性能を評価する．

分割 データセットを k 個に分割し，分割されたデータセットに 1 から k まで番号を付与する．分割されたデータセットはそれぞれ $\frac{10000}{k}$ 件ずつに分割される．

検証 1 分割されたデータセットのうち 1 番目のデータセットをテストデータに使用する．残りのデータを用いて機械学習モデルの学習，検証を行う．この時

のテストデータの精度 (Accuracy，またはその他の評価指標) を $result_1$ とする．

⋮

検証 k 分割されたデータセットのうち 1 番目のデータセットをテストデータに使用する．残りのデータを用いて機械学習モデルの学習，検証を行う．この時のテストデータの精度 (Accuracy，またはその他の評価指標) を $result_k$ とする．

結果

$$\frac{\sum_{i=1}^{k} result_i}{k} \tag{2.11.1}$$

を算出し，その結果を用いて精度の評価を行う．

以上のように評価を行うことで k 回の検証結果の平均値を得ることができる．そのため，手法の比較，機械学習モデルが安定して汎化性能を持っているかどうかなど，統計的な分析を行いたい場合によく用いられる方法である．本研究では，10-分割交差検証を用いることによって，学習済みモデルの未知データ（テストデータ）に対して特徴量抽出を行っている．

2.12 Batch Normalization

Batch Normalization とは，ニューラルネットワークモデルに使用されるアーキテクチャの一つである．まず，Batch と Normalization 分けて説明する．

Batch とは，機械学習モデル（特にニューラルネットワーク）を学習する際に利用されるミニバッチと呼ばれるものである．ニューラルネットワークモデルは計算量の膨大さが原因となり，データセットの全教師データに対して損失関数を計算してからパラメータを更新することができない場合がある．そこで，学習に使用するデータセットから教師データを何件かサンプリングし，あたかもサンプリングした教師データがデータセットの全教師データであるかのように学習を行う．このサンプリングした教師データの疑似データセットをミニバッチと呼ぶ，また，データセットから教師データを何件サンプリングするのかをバッチサイズと呼ぶ．

Normalization とは，分散が 1，平均が 0 になるように与えられたデータを変換

する方法である．日本語では標準化と呼ばれる．Normalization の式は与えらえられたデータ群を x(x はベクトルや行列), x の平均値を $\bar{x}$(x はスカラーやベクトル), x の標準偏差を S(S はスカラーやベクトル) とすると,

$$\frac{x-\bar{x}}{S} \tag{2.12.1}$$

で表される.

Batch Normalization はこれらの概念を統合したものであり，ニューラルネットワークの層と層の間に挿入され，前の層の出力に対して Normalization を行い次の層の入力に渡すというものである．Batch Normalization が発表されるまではデータセット内での Normalization が一般的であり，ミニバッチ内で Normalization は行われていなかった．Batch Normalization を導入することで学習が安定し，収束も早くなると言われている.

第 3 章　関連研究

1 章で述べた通り，教師あり学習に使用するデータセットに誤った教師データが存在するという問題があり，その問題に対するアプローチは主に 2 種類に分類できる．一つ目は教師データに誤りがあったとしても問題なく学習可能である，頑健な学習アルゴリズムを構築することである．二つ目は誤った教師データを検出し，削除または修正したデータセットを作成すること（データクレンジング）である．

頑健な学習アルゴリズムを構築する研究には，半教師あり SVM を用いた手法 [8] や，学習時にノイズをかけて学習を行うことで極端なノイズに強くなる学習アルゴリズム [9] などが存在する．例えば Eric らの研究 [10] では，損失関数を補正する手法を提案して頑健な学習方法を実現した．また Xia らの研究 [11] では，機械学習モデルのパラメータを精度向上に重要なパラメータとそうでないパラメータに分類し，分類結果に基づいて更新ルールを変更する手法を考案した．対して，本研究は誤った教師データが存在することを許さず，削除する方法を検討する研究であり，本研究は，二つ目のアプローチであるデータクレンジングの研究であるといえる．

データクレンジングの研究には，機械学習モデルの予測結果を用いる手法 [12]，k-NN を用いる手法 [13]，マルコフ確率場 (Markov Random Field) を用いる手法 [14] などが存在する．例えば，Vaibhav らの研究 [15] では，AQUAVS というオートエンコーダベースのニューラルネットワークモデルを使用する手法が考案されている．オートエンコーダは入力と出力が同じになるようにニューラルネットワークを学習させ，その中間層の出力を得ることで特徴量抽出器として利用可能である．ただし，分類問題を解いているわけではないため，教師データのクラスの情報を一切考慮しない特徴量となってしまう．そこで Vaibhav ら [15] は，オートエンコーダの学習時に同時に分類タスクも解かせることによって，教師ラベルのクラスごとに分かれた特徴量抽出器を作成した．その特徴量を用いて外れ値検出を行うことによって，データクレンジングを行っている．提案手法は Vaibhav らの研究 [15] と同様に，外れ値検出を用いるデータクレンジング手法である．

1 章でも述べたように，画像などの高次元データを扱う場合，外れ値検出手法に感度の低下，処理時間の増加というデメリットが発生する．これらの欠点を解決するために，特徴量抽出などを用いて次元削減を行い，外れ値検出を行う手法が提

案されている．外れ値検出手法には，SVM を用いた手法 [16, 17, 18]，ランダムフォレストを用いた手法 [19]，主成分分析を使用した手法 [20]，クラスタリング手法 [21, 22] を用いた手法，機械学習モデルのパラメータ更新に着目した手法 [23] などがある．我々の提案手法は機械学習モデルのパラメータ更新に着目した手法である．

提案手法と類似する手法として，Siqi らの研究 [23] がある．Siqi ら [23] は，inlier priority が提唱し，パラメータの更新方向に基づく $E^3Outlier$ という外れ値検出手法を提案した．inlier priority とは，outlier（本稿における誤った教師データ）は inlier（本稿における正しい教師データ）に比べて少数であるため，パラメータの更新は inlier の損失を下げることを優先するという主張である．この主張に基づいて考案された $E^3Outlier$ は，論文が発表された時点で画像の外れ値検出精度（AUROC）にて最高精度を記録した．inlier priority は，提案手法を考案する上でも重要な考え方である．詳細は 4 章で説明する．

$E^3Outlier$ と提案手法の共通点は機械学習モデルのパラメータに着目した点である．$E^3Outlier$ と提案手法の異なる点は二つある．一つ目はパラメータの特徴量を抽出する方法が異なる点である．$E^3Outlier$ は未学習の機械学習モデルを使用する．対して，提案手法は学習済みモデルを使用する．また，$E^3Outlier$ は入力時のパラメータの更新方向を特徴量とするため，パラメータの更新は一回である．対して，提案手法は入力データを分類できるようになった時点の重みを特徴量とするため，パラメータの更新は複数回である．二つ目は問題設定が異なる点である．$E^3Outlier$ の問題設定は，外れ値検出に教師データのうち，入力データのみを使用するという問題設定である．対して，提案手法の問題設定は，外れ値検出に教師データのうち，入力データと教師ラベルを使用するという問題設定である．つまり，$E^3Outlier$[23] は提案手法と同じ問題設定とは言えない．

提案手法と問題設定が同じ手法として，Confident Learning[3]，Label Fix[4] というデータクレンジング手法が存在する．これらの手法は問題設定が同じ手法の中で我々が知る限り最新の手法である．我々は問題設定が同じ手法で比較をするために，これら二つの手法との比較実験を行った．これらの手法と提案手法の共通点は問題設定の他に，学習済みモデルから得られた情報に基づいた手法という点もある．これら二つの手法の詳細な説明と，本研究との差分は 3.1 節，3.2 節で述べる．

3.1 Confident Learning

Confident Learning[3] は機械学習モデルの出力確率を利用した手法である．Confident Learning[3] の考え方は，データセット全体を見たときに入力データに教師ラベルが付与される確率とある入力データに対して人間が教師ラベルを付与したときに付与されている教師データを付ける確率の同時確率を利用することである．ある入力データ i に対して教師ラベル l が付与されていたとする．与えられたラベルを付与する確率は $p_g(l)$ と表す．人間が教師ラベルを付与する場合に l を付与する確率は $p_t(l)$ で表す．同時確率 $P(p_g(l), p_t(l))$ を計算することで，Confident Learning の考え方に基づいたスコアを算出できる．しかし，$p_t(l)$ を求めることはコスト面から現実的ではないため，$p_t(l)$ の代わりに学習済みモデルに i を入力したときの出力確率 $p_o(l)$ を使用する．$p_o(l)$ は学習済みモデルの出力に対して SoftMax 関数をかけた際の l の予測値に相当する値のことである．したがって，Confident Learning のスコアは同時確率 $P(p_g(l), p_o(l))$ の算出によって求められる．閾値を自動で設定する方法も提案されているが，6.2 節で行う比較実験ではこのスコアを使用して実験を行った．理由は 6.2 節で行う実験では，評価指標として閾値に依存しない AUROC を使用したためである．

Confident Learning[3] と提案手法の異なる点は，機械学習モデルから得る特徴量である．Confident Learning は機械学習モデルの出力値を使用する．対して，提案手法は機械学習モデルのパラメータを使用する．

3.2 Label Fix

Label Fix[4] は Confident Learning[3] と同様に機械学数モデルの出力確率を利用した手法である．ある入力データ i に対して教師ラベル l が付与されていたとする．学習済みモデルに i を入力して出力確率 $p_o(l)$ を求める．$p_o(l)$ は学習済みモデルの出力に対して SoftMax 関数をかけた際の l の予測値に相当する値のことである．$p_o(l)$ が確信度に相当するため，Label Fix のスコアは $p_o(l)$ の算出によって求められる．

Label Fix[4] と提案手法の異なる点は，Confident Learning[3] と同様に機械学習モデルから得る特徴量である．Label Fix は機械学習モデルの出力値を使用する．

対して，提案手法は機械学習モデルのパラメータを使用する．

第 4 章　提案手法

我々は，公開されているデータセットの中には，誤った教師データが存在することを問題視し，データクレンジング手法を提案する．本章では，提案手法に至った経緯，提案手法の考え方について詳しく説明する．4.1 節では，1 章で述べた仮説に至った経緯と，その考え方について述べる．4.2 節では，4.1 節で立てた仮説を基に提案手法を考案するまでの考え方について詳しく説明する．4.3 節では，4.2 節で述べた提案手法の特徴量抽出方法について詳しく述べる．4.4 節では，4.2 節で述べた提案手法のスコアの算出方法について詳しく述べる．

4.1　仮説のアイデア

我々は，データクレンジングを行うため外れ値検出に使用する特徴量の抽出方法を考えた．前提として，誤った教師データは正しいデータと比べて少数である．3 章でも述べたが，Wang ら [23] は inlier priority を提唱している．inlier priority は inlier（正しい教師データ）と outlier（誤った教師データ）の数には差があるため，機械学習モデルは inlier の損失を下げることを優先するというものである．この inlier priority を基に考えると，学習済みの分類境界は，正しい教師データの分類を行うために学習した境界であると言えると我々は考えた．すると，あるクラス a の分類境界外に存在する教師データのうち，教師ラベルがクラス a である教師データは誤りであると判断することができる．しかし，学習済みモデルは完璧に正しい判断ができるわけではない．その裏付けとして，Curtis らの研究 [3]，Nicolas らの研究 [4] がある．これらの研究では，学習済みモデルの出力が正しいという仮定の下で誤った教師ラベルを検出する手法が提案されている．しかし，これらの先行研究では，全ての誤った教師データを検出できていない．つまり，学習済みモデルは完璧に正しい判断ができるわけではないと言える．そのため，学習済みモデルの判断を基に，誤った教師データである可能性を示すような特徴量を作成する必要があると考えた．

そして我々は，複数のクラス $a, b, \cdots$ へデータを分類する問題を考えたとき，あるクラス a の分類境界外に存在する教師データのうち教師ラベルがクラス a であ

る教師データは，クラス a の分類境界から遠いほど誤った教師データである可能性が高いという仮説を立てた．基本的な考え方は 1 章で説明した通り，図 1.1 を用いて説明することができる．白$_1$ と 白$_2$ を比較したとき，白$_2$ の方が誤りである可能性が高いように見える．学習済みモデルでも同じ状態になっているのではないかと我々は考えた．

4.2 提案手法の考案

我々が立てた仮説に基づいて，我々はデータクレンジングを行うための手段を考えた．データクレンジングを行うために，外れ値検出を行う必要があり，そのための特徴量を抽出する手法が本研究の提案手法である．それ以外は既存の手法を用いている．

仮説を基に考えると，一番単純な方法は，誤った教師データである可能性がある教師データが，学習済みモデルの分類境界からどれだけ離れているかを調べることである．距離が近いならば，誤った教師データである可能性が高いとはいえず，距離が遠いならば，誤った教師データである可能性が高いといえる．しかし，学習済みモデルの分類境界を直接知る方法が存在しないという問題点がある．探索的な手法を用いることで，我々は学習済みモデルの分類境界を推定することが可能であるが，入力される可能性がある入力データをすべて用意する必要があり，処理時間の問題から現実的ではない．そこで我々は，学習済みモデルの分類境界の代替となる特徴量の抽出手法を考案することが必要であると考えた．

我々は，学習済みモデルの分類境界の代替となる特徴量はニューラルネットワークのパラメータであると考えた．学習済みモデルの分類境界は，訓練用データセットに含まれる入力データを付与されている教師ラベルであると分類できるように学習されたものである．ニューラルネットワークモデルの学習を行う際，学習の前後で変化するものはニューラルネットワークのパラメータ（重み，バイアス）である．以上のことから，ニューラルネットワークモデルの分類境界の学習は，ニューラルネットワークのパラメータの変化によって実現されていると我々は考えた．以上より我々は，外れ値検出のための特徴量として，ニューラルネットワークのパラメータを使用することを提案する．

4.3 特徴量抽出方法

4.2 節の考えに基づき，ニューラルネットワークのパラメータが分類境界を表しているという前提の下，外れ値検出のための特徴量抽出方法を考える．特徴量抽出において我々がこれまで述べたことから考慮しなければならない点が二つある．一つ目は，我々が立てた仮説には前提条件が存在することである．二つ目は，学習済みモデルの分類境界と誤った教師データの距離を表す特徴量を抽出することである．

まず，一つ目の考慮しなければならない点について述べる．4.1 節で述べた我々の仮説は，あるクラス a の分類境界が存在するという前提条件を内包している．そのため，まず最初に学習済みモデルを用意する必要がある．学習済みモデルは与えられたデータセットを学習することによって用意できる．学習済みモデルの学習方法の詳細は 5.1 節で説明する．

次に，二つ目の考慮しなければならない点について述べる．4.2 節で我々は分類境界の代わりとなる特徴量の抽出方法を提案した．しかし，本来我々が実現したい特徴量は，学習済みモデルの分類境界と教師データの距離を表す特徴量である．そこで，我々は用意した学習済みモデルをさらに追加で学習させ，分類境界を更新した後のパラメータを使用することを考えた．図 4.1 に分類境界と教師データの分布を表すイメージ図を示す．黒線は丸と四角のクラスの分類境界を表し，丸と四角は教師ラベルが異なる入力データであることを示している．用意した学習済みモデルは，図 4.1 の黒線に相当する分類境界を持っていると考える．この時の学習済みモデルのパラメータは黒線を表していると考えられる．図 4.1 の黒線から，分類境界外の教師データまでの距離を求めるためには，教師データを表す特徴量がなければならない．そこで我々は，学習済みモデルに対して，距離を求めたい教師データを入力し，分類可能になるまで追加で学習を行うこと（以下，追加学習と呼称）を考えた．すると，学習済みモデルの分類境界が変化し，距離を求めたい教師データを分類境界内に取り込むことになる．この時の追加学習を行った学習済みモデルのパラメータを使用することで，距離を求めたい教師データを含む分類境界の特徴量を得ることができる．そして我々は，学習済みモデルのパラメータと追加学習を行った学習済みモデルのパラメータの差を求めることによって，学習済みモデルの分類境界と教師データ間の距離を表す特徴量になると考えた．学習済みモデルの分類境界

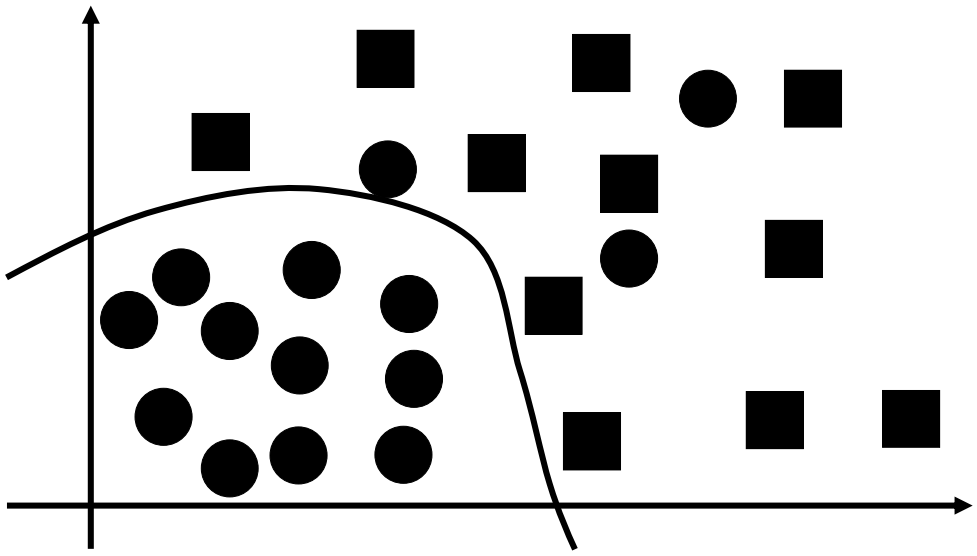

図 4.1　分類境界のイメージ図

と教師データ間の距離を表す特徴量の抽出方法についての詳細は 5.2 節で述べる．

4.4　スコアの算出

前節で述べた特徴量を抽出した後，データクレンジングを行うために外れ値検出を行う必要がある．そのためには，誤った教師データである可能性を表すスコアを算出する必要がある．そこで我々はマハラノビス距離を使用したスコアを算出することにした．

マハラノビス距離を使用する理由は，特徴量の散らばり方に応じた距離を算出することができ，パラメータの更新幅に影響されないと考えたためである．マハラノビス距離は，2.8 節で述べたように，与えられたデータ群の分散共分散行列を使用して算出可能な，データの散らばりを考慮した距離である．我々は，学習済みモデルを追加で学習する際，パラメータの更新幅は各重み，各バイアスごとに異なると考える．そのため，ユークリッド距離などを使用すると，更新幅の大きい重み，バイアスの影響が大きくなってしまうことが考えられる．マハラノビス距離はその点を解決した距離であるため，提案手法を用いた外れ値検出のスコアとして有用であ

ると考えた．

予備実験として，マハラノビス距離でスコアを算出した場合と，ユークリッド距離でスコアを算出した場合の比較実験を行った．その結果，マハラノビス距離でスコアを算出した場合の方が AUROC が高いことを確認した．よって我々の考えは正しいと考えられるため，我々はマハラノビス距離をスコアの算出に用いることにした．スコアの算出方法の詳細は 5.3 節で述べる．

第 5 章　実装

本章では，4 章で説明した提案手法を用いたデータクレンジング手法について，処理の手順を示しながら，数式を用いてより具体的な説明を行う．提案手法を用いた外れ値検出のスコア算出方法を Algorithm1 に示す．Algorithm1 で説明した変数と関数を以下で定義する．与えられたデータセットを D で表す．D は c クラス分類のデータセットである．D には N 件の教師データ

$$D = \begin{bmatrix} i_1 & s_1 & t_1 \\ \vdots & \vdots & \vdots \\ i_N & s_N & t_N \end{bmatrix} \tag{5.0.1}$$

が含まれており，$i_j(j = 1, 2, \cdots, N)$ は入力データ，$s_j(j = 1, 2, \cdots, N)$ は教師ラベル，$t_j(j = 1, 2, \cdots, N)$ は真の正解ラベルである．t_j は本章では利用しないが，6 章の説明で使用する．また，d_j は

$$d_j = \begin{bmatrix} i_j & s_j & t_j \end{bmatrix} \tag{5.0.2}$$

を指す．

k は $k-$ 分割交差検証用に用意する変数である．$k-$ 分割交差検証を行う理由については 5.2 節で述べる．

$Scores$ は外れ値検出を行うためのスコアを保存しておく変数である．$D.shape[0]$ は教師データの数を指している．Algorithm1 がすべて終了すると，6 行目で全て 0 に初期化した配列が算出されたスコアに置き換わる．$Scores[j]$ は d_j と対応している．そのため，$Scores[i]$ と s_j, t_j を比較することによって，外れ値検出の評価が可能である．

$D_{train}, D_{val}, D_{test}$ は k-分割交差検証をするために D を分割したデータセットである．$CrossValidation(dataset, k, itr)$ は k-分割交差検証用データセットを生成する関数である．与えられたデータセット $(dataset)$ を k 分割し，itr によって train データとして使用する箇所を変更する．Algorithm1 では，$k = 10$ と定義されているため，D を 10 個の小さなデータセットに分割した後で，8 個を train データ，残りの 2 個を validation データと test データに使用する．

Algorithm 1 提案手法の適用方法（スコアの算出まで）

$k \leftarrow 10$
$Scores \leftarrow [0] * D.shape[0]$
for $itr \leftarrow 1$ to k **do**
  $D_{train}, D_{val}, D_{test} \leftarrow CrossValidation(dataset = D, k = 10, itr = itr)$
  $M_b \leftarrow TrainModel(Model = M, train = D_{train}, val = D_{val})$
  $idx\_p \leftarrow [0] * D_{test}.shape[0]$
  $W \leftarrow [0] * D_{test}.shape[0]$
  $test\_cnt \leftarrow 0$
  for all $i_p, s_p, t_p \in D_{test}$ **do**
    $D_p \leftarrow [i_p, s_p, t_p]$
    $M_p \leftarrow M_b$
    while $Val\_Acc \neq 1.0$ **do**
      $M_p, ValAcc \leftarrow FTModel(Model = M_p, train = D_p, val = D_p)$
    end while
    $idx\_p[test\_cnt] \leftarrow p$
    $W[test\_cnt] \leftarrow GetWeights(Model = M_p)$
    $test\_cnt \leftarrow test\_cnt + 1$
  end for
  $mean \leftarrow Mean(W)$
  $cov\_inv \leftarrow GetCovInv(W)$
  $dists \leftarrow Mahalanobis(data = W, mean = mean, cov\_inv = cov\_inv)$
  $Scores\_tmp \leftarrow 1 - (dists/Max(dists))$
  for $i \leftarrow 0tolength(Scores\_tmp) - 1$ **do**
    $Scores[idx\_p[i]] \leftarrow Scores\_tmp[i]$
  end for
end for

M_b は学習済みモデルであり, Algorithm1 にて関数 $TrainModel(Model, tarain, val)$ に代入する M は学習前のニューラルネットワークモデルである．関数 $TrainModel(Model, train, val)$ はニューラルネットワークモデル (引数 $Model$), train データ (引数 $train$)，validation データ (引数 val) を入力として，M_b を構築する．$TrainModel(Model, train, val)$ で行われる処理の詳細は 5.1 節で説明する．

idx_p は D_{val} に含まれる教師データの D におけるインデックス情報を保存しておくための変数である．例えば，D_{val} の 5 番目のデータに d_{13} が含まれていたとすると，$idx\_p[5] \leftarrow 13$ が実行される．

W は取り出した一つの教師データによって追加学習が行われた M_b（Algorithm1 における M_p に相当）のパラメータを保存しておくための配列である．初期化は 0 で行われているが，実際に保存されるパラメータはベクトルである．ここで，関数 $GetWeights(Model)$ の動作について説明する．$GetWeights(Model)$ は与えらえたニューラルネットワークモデル (引数 $Model$) の最終層の重みを取得する関数である．例えば，M_b の出力層の一つ前のユニット数が u だとする．取得される重み（W に保存されるベクトル）の次元数は，ユニット間の全結合とバイアス項の結合を数えることで求められるため，$u \times c + c = (u + 1)c$ と算出できる．最終層の重みだけを用いる理由については 5.2 節で述べる．

$test\_cnt$ は特徴量抽出部分（Algorithm1 の 9 行目から 18 行目）のループ回数を数える変数である．この変数によって idx_p と W の同じインデックス番号に対応するデータを保存できるようにしている．

i_p, s_p, t_p は教師データの組である．この教師データは D を分割して作られた D_{test} に含まれる教師データである．下付き文字 p は D_{test} におけるインデックスではなく，D におけるインデックスを指している．idx_p に p を保存しておくことによって，Algorithm1 の 23 行目から 25 行目のループで $Scores$ の対応する箇所にスコアを保存できるようになっている．

D_p は一つの教師データによって追加学習を行うために取り出さされた D_{test} に含まれる教師データである．D_p は教師データを表す．詳細は 5.2 節で述べる．

$ValAcc$ は関数 $FTModel(Model, train, val)$ の返り値の一つであり，D_p の正解率を表す．関数 $FTModel(Model, train, val)$ は与えられたニューラルネットワー

クモデル (引数 $Model$) を訓練データ (引数 $train$) を一度学習することによって，M_p を更新する関数である．この時，M_p が i_p を s_p と分類できる場合，M_b と一緒に $ValAcc = 1.0$ を返し，分類できない場合は M_b と一緒に $ValAcc = 0.0$ を返す．

M_p は前述した通り，M_b に対して，D_p を用いて追加学習を行ったニューラルネットワークモデルである．M_p は 4.3 節で説明した特徴量を得るために必要なニューラルネットワークモデルである．

$mean$ は W の平均ベクトルである．平均値の計算は W 全体で行わず，ユニットの結合毎に独立して計算する．

cov_inv は W の分散共分散行列の逆行列である．$mean$ と cov_inv を求めることによってマハラノビス距離を算出することができる．

$dists$ はマハラノビス距離を計算した結果が保存されている変数である．関数 $Mahalanobis(data, mean, cov\_inv)$ は与えられたデータ (引数 $data$) の平均 (引数 $mean$)，分散共分散行列の逆行列 (引数 cov_inv) を使用してマハラノビス距離を計算する関数である．$dists$ は配列であり，例えば，W の 13 番目に対応する距離は $dist[13]$ に保存されている．

$Scores\_tmp$ は外れ値検出を行う際，閾値を超えるかどうかの判定に使用するスコアである．関数 $Max(dists)$ は与えられた配列の最大値を取得する関数である．$Scores\_tmp$ は配列であり，例えば，W の 13 番目に対応する距離は $dist[13]$ に保存されている．$Scores\_tmp$ は，1 に近いほど誤りである可能性が低く，0 に近いほど誤りである可能性が高いスコアである．

データセット D が与えられたとき，提案手法によるデータクレンジングで行われる処理は，以下に示す三つのフェーズに分けることができる．

データセットの学習　D を train データ，validation データ，test データに分割し，学習済みモデル M_b を構築する．（Algorithm1 の 5 行目）

特徴量抽出　M_b を一つの教師データを用いて追加学習を行い，その教師データを分類できるようになった時点のパラメータ情報を取得する．（Algorithm1 の 9 行目から 18 行目）

スコアリング　取得したパラメータを基にマハラノビス距離を用いたスコアを算出する．（Algorithm1 の 19 行目から 22 行目）

データセットの学習，特徴量抽出，スコアリングの各フェーズについての詳細は，それぞれ 5.1 節，5.2 節，5.3 節で述べる．

5.1 データセットの学習

4.3 節で述べた通り，4.1 節で述べた我々の仮説は，あるクラス a の分類境界が存在するという前提条件を内包している．そのため，提案手法を実現するためには学習済みモデルを用意する必要がある．そこで本節では，与えらえられたデータセット D を使用して学習済みモデル M_b を構築する方法について詳しく述べる．

D を $8:1:1$ の割合で D を train データ D_{train}，validation データ D_{val}，test データ D_{test} に分割する．ニューラルネットワークモデル M の訓練のために，以下の処理を繰り返す．

まず，ミニバッチを用意して訓練を行う．バッチサイズを m とすると，ミニバッチ B は，

$$B = \begin{bmatrix} i_{b1} & s_{b1} & t_{b1} \\ \vdots & \vdots & \vdots \\ i_{bm} & s_{bm} & t_{bm} \end{bmatrix} \tag{5.1.1}$$

で表される．このとき，下付き文字 $bm(m = 1, \cdots, m)$ は D_{train} に割り当てられた教師データからランダムに抽出された教師データの D におけるインデックス j である．次に M に i_b を入力し，M の出力 O を得る．O は，

$$O = \begin{bmatrix} o_{1,1} & \cdots & o_{1,c} \\ \vdots & \ddots & \vdots \\ o_{m,1} & \cdots & o_{m,c} \end{bmatrix} \tag{5.1.2}$$

と表される．そして，損失関数 $Loss(O, s_b)$ を用いて損失を計算し，最適化関数 $Optimizer(M, Loss(O, s_b))$ によって M の重み W，バイアス B を更新する．$Loss(O, s_b)$，$Optimizer(Loss(O, s_b), M)$ については，使用する関数によって式が異なるため，明記しない．また，最適化関数 $Optimizer(M, Loss(O, s_b))$ によって更新される M の重み W，バイアス B は，M が持つすべてのパラメー

タである必要はない．M の最終層など，一部のパラメータの更新（ファインチューニング）を行うだけでも問題はない．6 章の実験では損失関数に Pytorch*の $CrossEntropyLoss()$†，最適化関数に Pytorch の $Adam()$‡を使用している．

上記の処理を繰り返し，学習済みモデル M_b を構築する．上記の処理の内容は，Algorithm1 の関数 $TrainModel(Model, train, val)$ に相当する．ミニバッチは重複がないようにランダムに抽出される．これ以上抽出できなくなるまで繰り返したとき，e エポック目の学習が終了したという (e はこれ以上抽出できなくなるまで繰り返した回数)．e エポック目の学習が終了した後は，$e+1$ エポック目の学習が始まる．そのため，エポック数の上限を設定するなど，学習の終了条件を決める必要がある．

データセットの学習フェーズでは，エポックの上限回数を設定していない．その代わりに，D_val を用いて学習の終了条件を設定する EarlyStopping を使用している．EarlyStopping とは，validation データを用いて学習を監視することによって，過学習を防ぐ技術である．1 エポック終了するごとに，D_val を M に入力し，損失関数の値 l_e を得る．この時損失関数の値を保持しておき，設定したエポック数分の学習が終了しても損失関数の値が改善されなければ学習を停止する．損失関数の値が一番低かったエポック終了時の機械学習モデルを採用する．6 章の実験を例に説明する．6 章の実験では，EarlyStopping の条件を 10 エポックに設定した．50 エポック終了時に EarlyStopping の条件を満たしたとすると，損失関数の値が一番低いのは l_40 である．そのため，40 エポック終了時の M を M_b として採用する．プログラムを書く場合は，損失関数の最低値が更新されたら M を保存し，EarlyStopping の条件を満たしたら訓練のループから抜けるという処理を行う．

5.2 特徴量抽出

4.3 節で述べた通り， M_b の分類境界と，M_b の分類境界外の教師データまでの距離を求めるためには，教師データを表す特徴量がなければならない．我々は，M_b

*https://pytorch.org/

†https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

‡https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

の分類境界外の教師データの特徴量として，前節で構築した M_b に対して教師データ一つだけを入力して追加学習を行い，パラメータを特徴量として使用することを考えた．そこで本節では，特徴量の抽出方法について詳しく述べる．

M_b に対して，教師データを一つだけを入力して追加学習を行う．以降の説明は Algorithm1 の $FTModel(Model, train, val)$ にあたるこの教師データは D_{test} から取得する．D_{train}，D_{val} から取得した教師データは，学習時に使用されているため，過剰適合状態にある可能性がある．過剰適合状態とは，M の学習時に誤った教師データを学習したことによって，その誤った教師データを分類できる状態を指す．次に，D_{test} から取り出した一つの教師データ

$$D_p = \begin{bmatrix} i_p & s_p & t_p \end{bmatrix} \tag{5.2.1}$$

を M_b に学習させる．このとき，下付き文字 p は D のうち D_{test} に割り当てられた教師データからランダムに抽出された教師データのインデックス j である．M_b の学習方法は 5.1 節で述べた方法と同じである．ただし，使用するデータが一つであるため，ミニバッチの数式が以下のように変化する．

$$B = \begin{bmatrix} i_p & s_p & t_p \\ \vdots & \vdots & \vdots \\ i_p & s_p & t_p \end{bmatrix} \tag{5.2.2}$$

前節で説明したミニバッチとは異なり，同じデータがバッチサイズ分並んだだけのミニバッチになっている．M_b が i_p を s_p であると判定できるようになるまで学習を繰り返し，学習終了後の M_b を M_p と表す．このとき，学習は最終層のパラメータだけを更新する方法（ファインチューニング）を用いて行う．理由は次に述べる M_p から特徴量を得る方法の説明の中で述べる．

M_p から特徴量を得る方法について述べる (Algorithm1 の $GetWeights(Model)$)．M_p から重み，バイアスの情報を取得する．このとき，最終層のパラメータだけを取得する．最終層のパラメータとは，出力層とそのひとつ前の層の間に存在する全結合の重み，バイアスのことを指す．最終層のパラメータを抽出した特徴量を W と表記する．W はベクトルであり，その次元数は，出力層の一つ前の層のユニット数が u 個だったとき，W の次元数は $(u+1)c$ である．最終層のパラメータだけ

を取得する理由は二つある．一つ目の理由は全層のパラメータを使用すると次元数が増えるためである．二つ目の理由は最終層の重みのみを使用しても提案手法が適用可能なためである．

まず，一つ目の理由について詳しく述べる．ニューラルネットワークモデルの全パラメータを使用すると，入力画像よりも大きな次元数になりかねない．特徴量抽出を行う理由は，外れ値検出に使用する特徴量の次元数を減らすためである．そのため，次元数を減らすことが可能な，ニューラルネットワークの最終層のパラメータだけを使用することが適切であると考えた．

次に，二つ目の理由について詳しく述べる．我々は予備実験として，ロジスティック回帰モデルに提案手法を適用可能か調べた．その結果，ロジスティック回帰モデルに適用可能であることが分かった．ニューラルネットワークモデルの最終層の処理だけを考えると，ロジスティック回帰と同じ処理を行っている．そして，一つの教師データによって追加学習を行う際は，最終層のパラメータだけを更新しており，ロジスティック回帰と同じ処理をしている．以上のことから，我々は M_p の最終層のパラメータだけを抽出しても提案手法が適用可能であることが保証されていると考えた．

5.3 スコアリング

本節では，外れ値検出に使用するスコアの計算方法について述べる．4.4 節で述べたように，我々はマハラノビス距離を使用したスコアの計算を行う．スコアの計算方法はシンプルで，Algorithm1 の 19 行目から 22 行目に示された方法を用いてスコアの計算を行う．まず，マハラノビス距離を求めるためには，平均，分散共分散行列の逆行列を求める必要がある．その次に，マハラノビス距離を計算し，$Scores\_tmp = 1 - (dists/Max(dists))$ で求めることができるスコアを算出する．このスコアは，1 に近いほど誤りである可能性が低く，0 に近いほど誤りである可能性が高いスコアである．計算したスコアは D_{test} の教師データのみなので，D に対応するように $Scores$ に保存しておく必要がある．D と対応がとれるように，$Scores$ にスコアを保存している部分が Algorithm1 の 23 行目から 25 行目である．

4 章では，M_b と M_p の差分を特徴量とするという記述があったが，$Scores\_tmp$

の計算では，M_b が出てきていない．その理由について説明する．マハラノビス距離はデータの平均座標からの距離が等高線のようになっている距離であり，その距離はデータの分布に基づいて計算される．つまり，M_p の特徴量を使用した場合のマハラノビス距離は，M_p の平均点からの距離を求めることになる．一方，M_p の特徴量から M_b の特徴量を引いた特徴量を使用した場合のマハラノビス距離は，平均座標がずれるものの，平均座標からの距離とデータの分布は変わらない．M_b の特徴量という定数を引いているにすぎず，データの分布が平行移動するだけである．以上のことから，M_b の特徴量を引くという処理は無駄な工程であると考えることができるため，実装では省いている．

第 6 章　評価実験

本章では，我々が行った実験の結果とその考察について述べる．6.1 節では，本章の実験で行う実験に使用するデータセットについて説明する．6.2 節では，Confident Learning[3]，Label Fix[4] と提案手法を用いたデータクレンジングの精度比較実験の結果を報告する．6.3 節では，6.2 節で判明した提案手法の問題点について，その原因の調査，解決策の考案を行う．6.4 節では，6.3 節で考案した提案手法の問題点の解決策によって，問題点が解消されたかどうかを調査する．提案手法，比較手法の精度は AUROC を用いて評価を行った．

6.1　使用データセット

本章の実験では, MNIST*, FashionMNIST†, SVHN‡, CIFAR10§, CIFAR100¶を使用した．本節では，これらのデータセットの内容について詳しく述べる．

MNSIT は 70,000 件の教師データを含む画像分類用データセットである．MNIST が持つ教師データの入力データと教師ラベルについて説明する．MNIST の入力データは手書きの数字をモノクロにした画像である．入力データはモノクロ画像であり，画像サイズは 28×28 の正方形の画像である．教師ラベルの種類は 10 種類である．MNIST の 10 種類のラベルは，入力データが 0 から 9 までの数字のどの数字のデータなのかを表しており，教師ラベル 0 から 9 は手書き文字の 0 から 9 に対応している．

FashionMNIST は 70,000 件の教師データを含む画像分類用データセットである．FashionMNIST が持つ教師データの入力データと教師ラベルについて説明する．FashionMNIST の入力データは服の画像をモノクロにした画像である．入力データはモノクロ画像であり，画像サイズは 28×28 の正方形の画像である．教師ラベルの種類は 10 種類である．FashionMNIST の 10 種類のラベルは，入力デー

*http://yann.lecun.com/exdb/mnist/
†https://github.com/zalandoresearch/fashion-mnist
‡http://ufldl.stanford.edu/housenumbers/
§https://www.cs.toronto.edu/~kriz/cifar.html
¶https://www.cs.toronto.edu/~kriz/cifar.html

タがどの種類のファッション商品の画像データなのかを表している．教師ラベル 0 から 9 は

ラベル 0： T シャツ，トップス
ラベル 1： ズボン
ラベル 2： プルオーバー
ラベル 3： ドレス
ラベル 4： コート
ラベル 5： サンダル
ラベル 6： シャツ
ラベル 7： スニーカー
ラベル 8： バッグ
ラベル 9： アンクルブーツ

に対応している．

SVHN は 99,289 件の画像分類用データセットである．SVHN が持つ教師データの入力データと教師ラベルについて説明する．SVHN の入力データは，現実世界に存在する数字の画像である．例えば，店の看板に記載された店の電話番号の画像が含まれている．入力データはカラー画像であり，画像サイズは 32×32 の正方形の画像である．教師ラベルの種類は 10 種類である．画像の中心に写っている数字が教師ラベルになっており，MNIST と同様に 0 から 9 までの数字に対応したラベルが付与されている．

CIFAR10 は 60,000 件の画像分類用データセットである．これらのデータセットが持つ教師データの入力データと教師ラベルについて説明する．CIFAR10 の入力データは乗り物や動物など 10 種類のカラー画像である．入力データは SVHN と同様にカラー画像であり，画像サイズは 32×32 の正方形の画像である．教師ラベルの種類は 10 種類である．CIFAR10 の 10 種類のラベルは，入力データがどの種類のカラー画像なのかを表している．教師ラベル 0 から 9 は

ラベル 0： 飛行機
ラベル 1： 自動車
ラベル 2： 鳥

ラベル 3： 猫
ラベル 4： 鹿
ラベル 5： 犬
ラベル 6： カエル
ラベル 7： 馬
ラベル 8： 船
ラベル 9： トラック

に対応している．

CIFAR100 は 60,000 件の画像分類用データセットである．これらのデータセットが持つ教師データの入力データと教師ラベルについて説明する．CIFAR100 の入力データは乗り物や動物など 100 種類のカラー画像である．入力データは SVHN や CIFAR10 と同様にカラー画像であり，画像サイズは 32×32 の正方形の画像である．教師ラベルの種類は 100 種類であるが，本章の実験ではその上位クラスとなる 20 種類のラベルを使用した．CIFAR100 の 100 種類のラベルについては，本稿で使用していないため，説明を省く．CIFAR100 の 20 種類の入力データがどの種類のカラー画像なのかを表している．教師ラベル 0 から 19 は

ラベル 0： 水生動物（イルカやクジラなど）
ラベル 1： 魚
ラベル 2： 花
ラベル 3： 食品容器
ラベル 4： 果物と野菜
ラベル 5： 家電
ラベル 6： 家具
ラベル 7： 昆虫
ラベル 8： 大型の肉食動物
ラベル 9： 大型の建造物
ラベル 10： 大自然の風景
ラベル 11： 大型の雑食動物と草食動物
ラベル 12： 中型の哺乳類

ラベル 13： 昆虫ではない無脊椎動物
ラベル 14： 人
ラベル 15： 爬虫類
ラベル 16： 小型の哺乳類
ラベル 17： 木
ラベル 18： 車両 1（バイクと車と電車）
ラベル 19： 車両 2（特殊車両，路面電車，ロケットなど）

に対応している．

6.2 実験 1：比較実験

本節では，Confident Learning[3]，Label Fix[4] と提案手法を用いたデータクレンジングの精度比較実験の結果と考察を報告する．本節で行う実験の目的は，従来のデータクレンジング手法と比較して提案手法の優位性を示すことである．6.2.1 節では，実験設定について説明する．6.2.2 節では，実験の結果を報告し，その結果に対する考察を 6.2.3 節で行う．提案手法の適用範囲については，7 章で述べる．

6.2.1 実験設定

まず本章で行う実験について，共通する実験設定を先に述べ，そのあとに本節固有の設定について説明する．本章で共通する実験設定は，損失関数，最適化関数，データセットの学習（M_b の構築）時の EarlyStopping，バッチサイズ，追加学習の上限エポック数，ダミーデータセットの生成方法，評価指標の設定である．ダミーデータセットとは，データセットに存在する教師データの教師ラベルを別のラベルに変更し，わざと誤った教師データを作ったデータセットの事である．本節固有の実験設定は，使用データセット，ダミーデータセット生成用パラメータ，使用ニューラルネットワークモデルである．

まず，共通する実験設定について述べる．本章で行う実験の損失関数，最適化関数，EarlyStopping，バッチサイズ，追加学習の最大エポック数の設定は以下に示す条件である．

損失関数： Pytorch[‖]の $CrossEntropyLoss()$[**]のデフォルトを使用.

最適化関数： Pytorch の $Adam()$[††]のデフォルトを使用.

EarlyStopping： 損失関数が 10 エポック更新されなければ終了.

バッチサイズ： 32 に設定.

追加学習の最大エポック数： 100 に設定し，終了条件を満たせなかった場合は誤った教師データであると判断する.

損失関数，最適化関数は 5.1 節で説明したデータセットの学習と，5.2 節で説明した一つの教師データのみを使用した追加学習のどちらも同じ設定である.

ダミーデータセットの生成方法は，比較実験で使用する Confident Learning[3] の論文の実験方法に準拠している．Confident Learning と本研究のどちらの実験においても，使用するデータセット（MNIST や CIFAR10 など）はすべて真に正しい教師データであると仮定し，一部の教師ラベルを別のラベルに変更することで誤った教師データを作成する．また，Confident Learning の実験では，ランダムに誤った教師データを生成すると，実世界の教師データの誤りの分布と乖離が起きてしまうと主張されている．そのため，Confident Learning の実験では，ミスラベル率 $\rho(0 \leq \rho \leq 1)$ とスパース性 $\sigma(0 \leq \sigma \leq 1)$ を使用して, 実世界の教師データの誤りに近い誤った教師データを含むダミーのデータセットを生成している．ミスラベル率 ρ は誤った教師データを生成する割合を表している．例えば，ミスラベル率を 0 に設定すると, 生成されるダミーデータセットは全て正しい教師データで構成され, ミスラベル率を 1 に設定すると生成されるダミーデータセットは全て誤った教師データで構成されることになる．スパース性 σ はどの程度偏って教師ラベルを誤るかを表している．例えば，スパース性 σ を 0 に設定すると，生成される誤った教師データは, 元々付与されていた教師ラベルではない教師ラベルがランダムで付与される．σ を 1 に設定すると，生成される誤った教師データは，元々付与されていた教師ラベルではない一つの教師ラベルに偏って付与される．このミスラベル率とスパース性に基づいたダミーデータセットの生成は Confident Learning の論文にて python

[‖] https://pytorch.org/

[**] https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

[††] https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

モジュール（Cleanlab[‡‡]）が公開されている．Cleanlab にて公開されているモジュールのうち，我々は noise_generation.generate_noise_matrix_from_trace モジュールと noise_generation.generate_noisy_labels モジュールを使用してダミーデータセットを生成した．これらのモジュールの動作については Cleanlab のリファレンス[§§]を参照されたい．

我々は評価指標に AUROC を用いた．AUROC は，閾値を連続的に変更したときの真陽性率と偽陽性率によって描かれる曲線の下側の面積を用いて評価する評価指標である．真陽性率と偽陽性率の求め方について説明する．5 章で説明した $i_j(j = 1, 2, \cdots, N)$ はデータセットの入力データ，$s_j(j = 1, 2, \cdots, N)$ はダミーデータセット作成後の教師ラベル，$t_j(j = 1, 2, \cdots, N)$ はダミーデータセット作成前の教師ラベルに相当する．$s_j(j = 1, 2, \cdots, N)$ と $t_j(j = 1, 2, \cdots, N)$ について $s_j = t_j$ ならば True(正しい教師データ) であり，$s_j \neq t_j$ ならば False（誤った教師データ）であると判定する．我々は，提案手法の予測結果と上記の判定結果が一致したかどうかによって真陽性率と偽陽性率を求める．詳細は 2.9 節を参照されたい．閾値を連続的に変更したときの真陽性率と偽陽性率によって描かれる曲線は ROC 曲線 (Receiver Operatorating Characteristic curve) と呼ばれる．曲線の下側の面積は AUC(Area Under the Curve) と呼ばれる．これらを合わせて AUROC と呼ぶ．AUROC は真陽性率と偽陽性率の値域が 0 から 1 であるため，AUROC の値域も 0 から 1 である．AUROC は 1 に近いほど性能が高く，0.5 に近づくとランダム予測と変わらない精度だと判断できる評価指標である．AUROC は閾値を必要としない評価指標であり，本稿で我々が提案する手法には閾値を決める方法が含まれていないため，AUROC を評価指標として採用した．

次に本節固有の実験設定について述べる．使用データセットは MNIST，FashionMNIST，SVHN，CIFAR10，CIFAR100 である．ダミーデータセット生成用パラメータ（ρ と σ）は，ρ を [0.2, 0.4]，σ を [0, 0.2, 0.4, 0.6] で変化させる．そのため，本節では計 8 パターンの実験を行った結果を報告する．使用したニューラルネット―ワークモデルは図 6.1 に示したモデルである．図 6.1 に示した四角で囲まれたニューラルネットワークの構成要素について述べる．$Conv2d(1, 6, 5, 1, 1)$ は，

[‡‡] https://cleanlab.ai/

[§§] https://docs.cleanlab.ai/stable/cleanlab/benchmarking/noise_generation.html

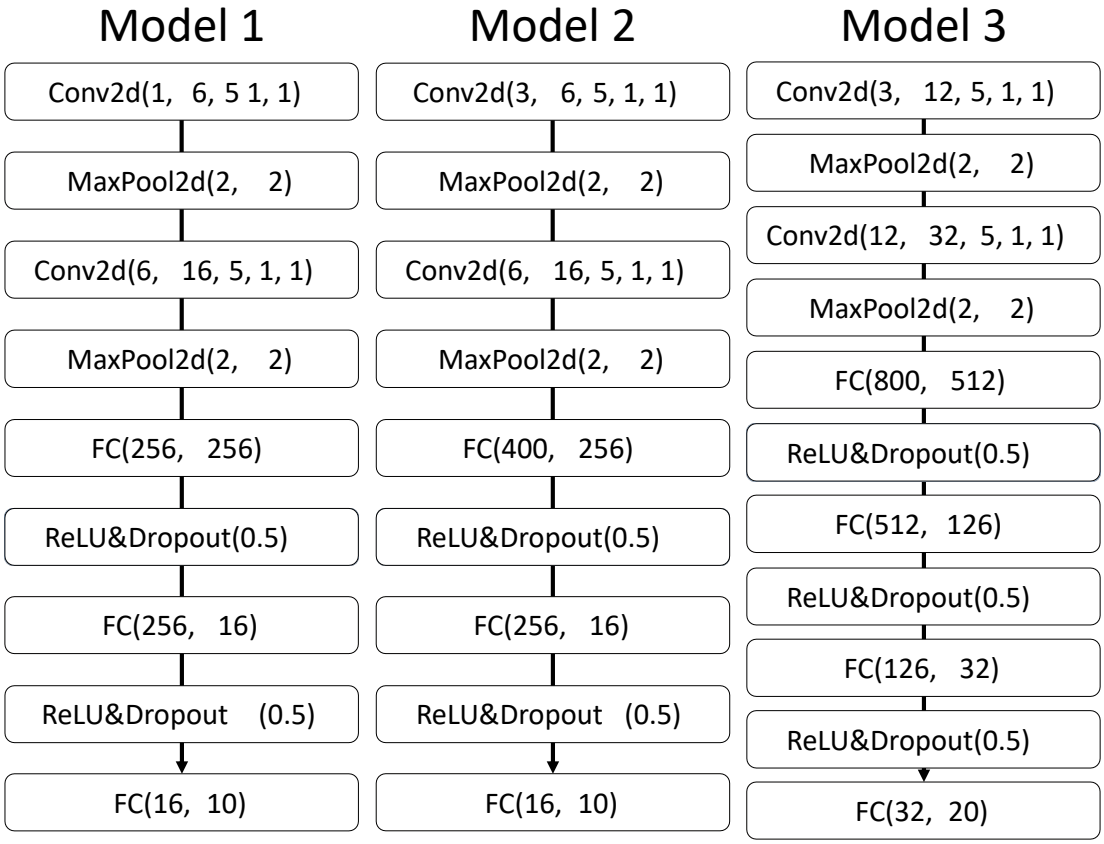

図 6.1　実験 1 で使用したニューラルネットワークモデル

畳み込み層を表している．左から順に，入力チャンネル数，出力チャンネル，カーネルサイズ，ストライド，パディングの設定の値を示している．$MaxPool(2,2)$ はカーネルサイズを 2，ストライドを 2 に設定した MaxPooling を行うプーリング層を表している．$FC(256,256)$ は入力ベクトル 256 次元，出力ベクトル 256 次元の全結合層を表している．$ReLU\&Dropout(0.5)$ は活性化関数として ReLU 関数を使用し，0.5 の確率で出力を 0 にするドロップアウトを適用していることを表している．使用したデータセットのうち MNIST，FashionMNIST には Model 1 のニューラルネットワークモデルを使用した．SVHN，CIFAR10 には Model 2 のニューラルネットワークモデルを使用した．CIFAR100 には，Model 3 のニューラ

ルネットワークモデルを使用した.

6.2.2 結果

提案手法と Confident Learning[3], Label Fix[4] の比較実験を行った結果を表 6.1 に示す. ρ, σ はそれぞれミスラベル率とスパース性のパラメータを指しており, 比較手法 (Confident Learning, Label Fix) と提案手法 (our method) の名前の横に各条件における各手法の AUROC(%) の平均値を示している. 各データセット名の下に記載されている結果がそのデータセットを使用した際の結果を表してい

表 6.1 比較実験の結果

ρ	0.2				0.4			
σ	0	0.2	0.4	0.6	0	0.2	0.4	0.6
	MNIST							
Confident Learning	95.30	91.49	88.96	82.62	**99.57**	**99.59**	**99.48**	**99.16**
Label Fix	**96.40**	91.15	79.83	70.12	99.28	96.07	90.83	80.54
our method	95.79	**93.93**	**90.88**	**85.26**	99.54	99.46	99.26	98.76
	FashionMNIST							
Confident Learning	91.84	88.17	85.46	79.73	96.59	96.48	96.31	**94.51**
Label Fix	**93.38**	88.66	78.54	67.56	97.02	94.17	89.53	77.51
our method	93.21	**91.22**	**87.69**	**82.36**	**97.47**	**97.04**	**96.61**	94.49
	SVHN							
Confident Learning	90.95	88.90	86.16	81.07	**96.52**	**96.43**	**95.79**	**94.46**
Label Fix	91.79	87.67	78.76	69.28	94.84	92.31	86.75	77.03
our method	**92.20**	**90.71**	**87.60**	**83.01**	96.45	96.18	95.54	93.74
	CIFAR10							
Confident Learning	77.99	76.21	73.34	70.15	85.11	84.92	83.48	81.40
Label Fix	81.76	78.60	72.53	65.22	87.08	84.68	81.51	73.24
our method	**81.90**	**80.28**	**76.55**	**71.91**	**88.03**	**87.48**	**85.26**	**81.99**
	CIFAR100							
Confident Learning	63.96	65.53	65.40	61.94	78.38	78.09	77.89	76.85
Label Fix	**80.95**	77.22	72.81	68.35	84.28	82.32	79.66	74.26
our method	80.24	**77.81**	**74.70**	**70.41**	**84.83**	**83.72**	**82.62**	**79.96**

る．AUROC(%) の値は，データセット，条件ごとに一番 AUROC の値が高いものを太字表記にしており，その中でも統計的に有意な差がある AUROC の値は下線が引かれている．

得られた結果からわかる事実は以下の三つである．

- 条件によって提案手法が良いといえる結果と，そうでない結果が得られた．
- 提案手法の ARUOC が比較した中で最高値だった回数は全 40 条件のうち 28 条件 (70%) である．
- 提案手法の AUROC が比較した中で最高値であり，統計的な有意差があった回数は全 40 条件のうち 8 条件 (20%) である．

次節では，この結果を踏まえた考察を行う．

6.2.3 考察

本節で我々が議論したいことは二つある．一つ目は，提案手法が比較手法よりも優れていると主張する根拠についてである．二つ目は，なぜ提案手法が比較手法よりも優れているのかについてである．

まず，提案手法が比較手法よりも優れていることを 6.2.2 節で得られた結果から示す．6.2.2 節でも述べた通り，提案手法の勝率は 70% であり，統計的な有意差を示した条件が 20% 存在する．対して Confident Learning の結果は, 勝率が 25.5%, 有意差があったのは 7.5% である．また，Label Fix の結果は，勝率が 7.5%, 有意差があったのは 0% である．以上のことから我々は提案手法が最も優れた手法であると主張する．

我々は，学習済みモデルを用いてテストデータを分類した場合の Accuracy が低いデータセットを使用してデータクレンジングを行うほど，上記の主張がより強固になると主張する．図 6.2 にミスラベル率の設定毎に M_b を用いて test データを予測した際の Accuracy（正解率）の平均値を求めた結果を示す．図 6.2 の縦軸は M_b を用いて test データを予測した際の Accuracy（正解率），横軸はデータセット，$\rho = 0.4, 0.2$ はミスラベル率を表している．図 6.2 から，我々が用意したデータセットの難易度（難しいほど大きい）は $MNIST \leq FashionMNIST \leq SVHN \leq$

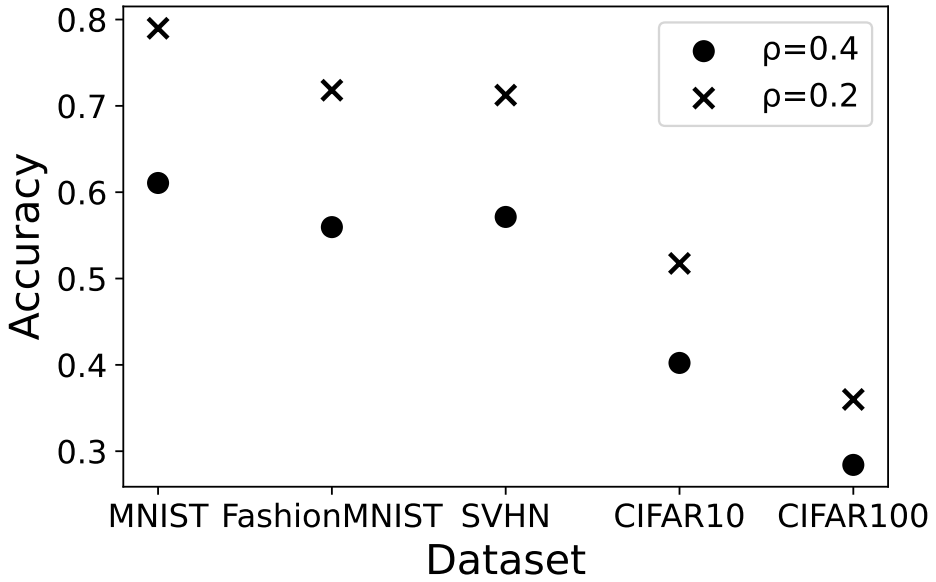

図 6.2　ミスラベル率とデータセット毎の Accuracy

$CIFAR10 \leq CIFAR100$ の関係があると我々は考えた．表 6.1 を見ると，データセットの難易度が上がるにつれて提案手法の勝率，有意差を示した回数が増えていくことが確認できる．以上のことから，我々の主張は特に難易度が高いデータセットであればあるほど強固なものになると考える．

我々は，このような結果が得られた原因は簡単なデータセットを用いると比較手法間で差が出ないためであると考えた．まず，図 6.2 を見ると，MNIST，FashionMNIST，SVHN の Accuracy はミスラベル率が 0.2 の場合は 0.8 から 0.7，ミスラベル率が 0.4 の場合は 0.6 付近の値を取っている．これらの Accuracy の値は，正しい教師データの割合に近い値であるといえる．ダミーデータセットの誤った教師データを誤分類している可能性もあるが，inlier priority[23] を考慮すると，正解した教師データの大部分は正しい教師データであると考えられる．ここまでの考えが正しいとすると，学習済みモデルはすでに正しいデータと誤った教師データを判別する能力を備えていると考えることができる．また，提案手法，Confident Learning，Label Fix はそれぞれ，学習済みモデルから得られる特徴量を用いた手

法である．そのため，それぞれの手法に大きな差が出なかったという可能性が考えられる．

次に，なぜ提案手法が比較手法よりも優れているのかについて議論する．我々は，提案手法によって AUROC が向上した理由は，比較手法と比べて特徴量が多いことと，その特徴量が優れていることが要因だと考える．提案手法と比較手法を比べたとき，提案手法が優れている理由になり得る提案手法の特徴は以下に示す二点である．一つ目は，比較手法は学習済みモデルの出力値を使用したのに対して，提案手法では学習済みモデルのパラメータを使用している点である．二つ目は，比較手法と比べて提案手法の方が特徴量数が多い点である．

学習済みモデルの出力値を利用するか，パラメータを使用するかの違いが精度向上の要因であると考える場合について述べる．AUROC が向上した理由は，学習済みモデルの出力という特徴量よりも学習済みモデルのパラメータという特徴量法が優れた特徴量であることが考えられる．特に，提案手法は学習済みモデルのパラメータをさらに一つの教師データに特化したパラメータに変換している．そのため，学習済みモデルをただ使用するよりもその教師データの特徴をうまく表現できている可能性があると考えられる．

比較手法と比べて提案手法の方が特徴量数が多い点が精度向上の要因であると考える場合について述べる．AUROC が向上した理由は，特徴量の次元数が増加することによって外れ値検出に必要な情報が得られるようになったためであると考えられる．機械学習において，精度の向上を図るために必要な要素はデータ数と特徴量の次元数だと言われている．外れ値検出においては 1 章で述べた次元の呪いという考え方もあるが，それ以上に特徴量の次元数の向上が外れ値検出に良い影響を与えたと考えられる．

6.3　実験 2：他のニューラルネットワークへの適用

6.2 節では，提案手法が比較手法よりも優れていることを示すための実験を行った．本節では，公開されているニューラルネットワークモデルを使用しても提案手法が適用可能かどうかを調査し，その精度 (AUROC) について評価する．本実験の目的は，提案手法が自作したニューラルネットワークモデルだけではなく，どの

ようなニューラルネットワークモデルでも使用可能なことを示すことである．6.3.1 節では，実験設定について説明する．6.3.2 節では，実験の結果を報告し，その結果に対する考察を 6.3.3 節で行う．6.3.4 節では，6.3.3 節で判明した提案手法の問題点を解決するための方法を説明する．

6.3.1 実験設定

本節固有の実験設定（使用データセット，ダミーデータセット生成用パラメータ，使用ニューラルネットワークモデル）について述べる．本節固有ではない実験設定については 6.2.1 節を参照されたい．

使用したニューラルネットワークモデルは，図 6.1 の Model 2，AlexNet[24]，Vgg-16[25]，ResNet-18[6]，DenceNet-121[26]，EfficientNet-b7[27] の 6 種類のモデルである．使用したデータセットは SVHN のみであり，ダミーデータセット生成用パラメータはミスラベル率 0.25，スパース性を 0 に設定した．

6.3.2 結果

実験の結果を図 6.3 に示す．図 6.3 の縦軸は AUROC，横軸は使用したモデルの名称を表している．

図 6.3 からわかるように，Model 2，AlexNet，Vgg16 では AUORC が高く，ResNet-18，DenseNet-121，EfficientNet-b7 では AUROC がランダム予測と同程度まで低下した．提案手法はニューラルネットワークモデルに適用可能な手法である一方で，ニューラルネットワークモデルによって精度にばらつきがあることがわかった．

6.3.3 考察

6.3.2 節で得られた結果から，提案手法はニューラルネットワークモデルに適用可能であるが，AUROC が低下するニューラルネットワークモデルが存在することが分かった．AUROC が低下するニューラルネットワークを用いた場合，提案手法の

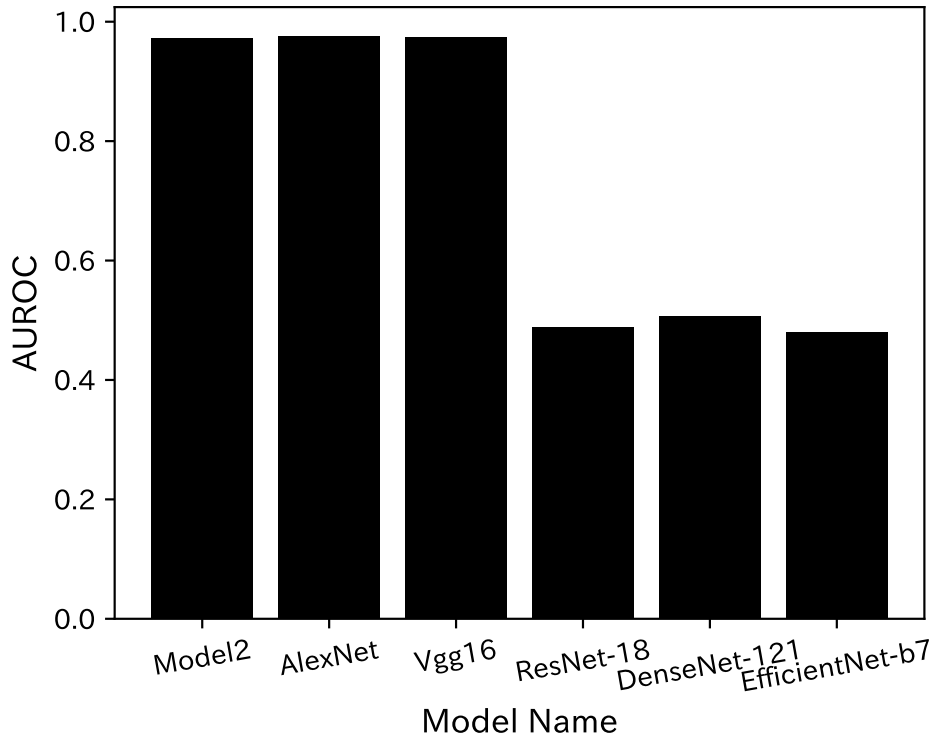

図 6.3　使用したニューラルネットワークモデルと AUROC の関係

精度はランダム予測と変わらない結果となった．我々は，使用するニューラルネットワークモデルによって，AUROC の低下が起きることは問題であると考えた．そこで我々は，AUROC が 6.2 節で得られた結果と同程度のニューラルネットワーク

表 6.2　AUROC 低下の原因となり得る要素と各ニューラルネットワークモデルの関係

モデル	Model2	AlexNet	Vgg16	ResNet-18	DenseNet-121	EfficientNet-b7
パラメータ数	109,810	61,117,026	138,373,730	11,705,698	7,995,042	63,828,106
Batch Normalization	×	×	×	○	○	○
Stochastic Depth	×	×	×	×	×	○
スキップコネクト	×	×	×	○	○	○
残差ブロック	×	×	×	○	×	×
Dense ブロック	×	×	×	×	○	×
MBCConv ブロック	×	×	×	×	×	○

モデルと AUROC が低下するニューラルネットワークモデルを比較し，原因を調査することによってこの問題を解決しようと考えた．

我々は，各ニューラルネットワークモデル（図 6.1 の Model 2，AlexNet[24]，Vgg-16[25]，ResNet-18[6]，DenceNet-121[26]，EfficientNet-b7[27]）間におけるニューラルネットワークの構造の違いを表 6.2 にまとめた．我々は，パラメータ数，Batch Normalization 使用の有無，Stochastic Depth 使用の有無，スキップコネクト使用の有無，残差ブロック使用の有無，Dense ブロック使用の有無，MBCConv ブロック使用の有無の計 7 種類のニューラルネットワークの構造の違いを挙げた．表 6.2 は，それらのニューラルネットワークの構造について，パラメータ数はパラメータの総数，それ以外のニューラルネットワークの構造は，使用する場合は○，使用しない場合は×を表示している．これにより，ニューラルネットワークモデルとの関係を示している．

図 6.3 から AUROC が低下するモデルは，ResNet-18[6]，DenceNet-121[26]，EfficientNet-b7[27] であることがわかる．そのことを踏まえて表 6.2 を見ると，AUROC の低下が起こる原因は Batch Normalization もしくはスキップコネクトのどちらかであると考えられる．

次に我々は，原因と考えられる二つのアーキテクチャの中でどちらが AUROC が低下する原因になり得るかを考えた．そして我々は Batch Normalization が原因であると結論付けた．我々は Batch Normalization が一つの教師データのみを入力とすることによって意図しない挙動を起こし，結果として提案手法の考えに基づいた手法の実現ができていないことを発見した．なぜなら，5.1 節で説明したデータセットの学習フェーズと，5.2 節で説明した一つの教師データを用いた追加学習フェーズでは，ミニバッチに含まれる教師データの分布にずれが生じてしまうためである．

Batch Normalization は，ミニバッチが与えられたとき，そのバッチ内のデータ全体を標準化するものである．我々は，5.1 節でミニバッチは

$$B = \begin{bmatrix} i_{b1} & s_{b1} & t_{b1} \\ \vdots & \vdots & \vdots \\ i_{bm} & s_{bm} & t_{bm} \end{bmatrix} \tag{6.3.1}$$

と表されると説明し，5.2 節では，一つの教師データのみを使用するため，

$$B = \begin{bmatrix} i_p & s_p & t_p \\ \vdots & \vdots & \vdots \\ i_p & s_p & t_p \end{bmatrix} \tag{6.3.2}$$

と表すと説明した．式 6.3.1 と式 6.3.2 のミニバッチの違いは，train データ D_{train} からランダムに抽出された複数の教師データか，test データ D_{test} からランダムに抽出された一つの教師データを複製したものかという違いがある．この違いを細分化すると，train データを使用するか，test データを使用するかという違いと，複数の教師データを使用するか，一つの教師データを使用するかという違いに分けることができる．Batch Normalization の意図しない挙動について，我々はこれらの違いのうち後者が原因であると考える．

我々は複数の教師データを使用する場合と，一つの教師データを使用する場合の違いについて考えた．複数の教師データを使用する場合，ミニバッチは複数種類の教師ラベルと複数種類の入力データを持つ教師データの集合であるといえる．対して，一つの教師データを使用する場合，ミニバッチは単一の教師ラベルと単一の入力データを持つ教師データの集合であるといえる．以上に示したように我々は，5.1 節で説明したデータセットの学習フェーズと，5.2 節で説明した一つの教師データを用いた追加学習フェーズでは，ミニバッチに含まれる教師データの分布に違いが生じていると考えた．そして，Batch Normalization を含むニューラルネットワークモデルを使用すると，その違いが生じたまま標準化を行うことになる．そのため，標準化された入力データはデータセット学習時と追加学習時で値が全く異なるといえる．提案手法は，学習済みモデルの分類境界からどの程度離れているかを特徴量化するものである．データセット学習時と追加学習時で入力データの値が全く異なる場合，学習済みモデルの分類境界が意味をなさない．つまり，提案手法の考え方に基づいた手法になっていないといえる．よって我々は，AUROC が低下した原因は Batch Normalization であると判断した．

AUROC が低下する原因が Batch Normalization であることを示すために，全ての畳み込み層の後に Batch Normalization を入れた場合と入れなかった場合の比較を行った．使用データセットは SVHN と CIFAR10，ミスラベル率 ρ を 0.8，スパース性 σ を [0.0, 0.4] に設定した場合の計 4 パターンの比較を行った．使用モ

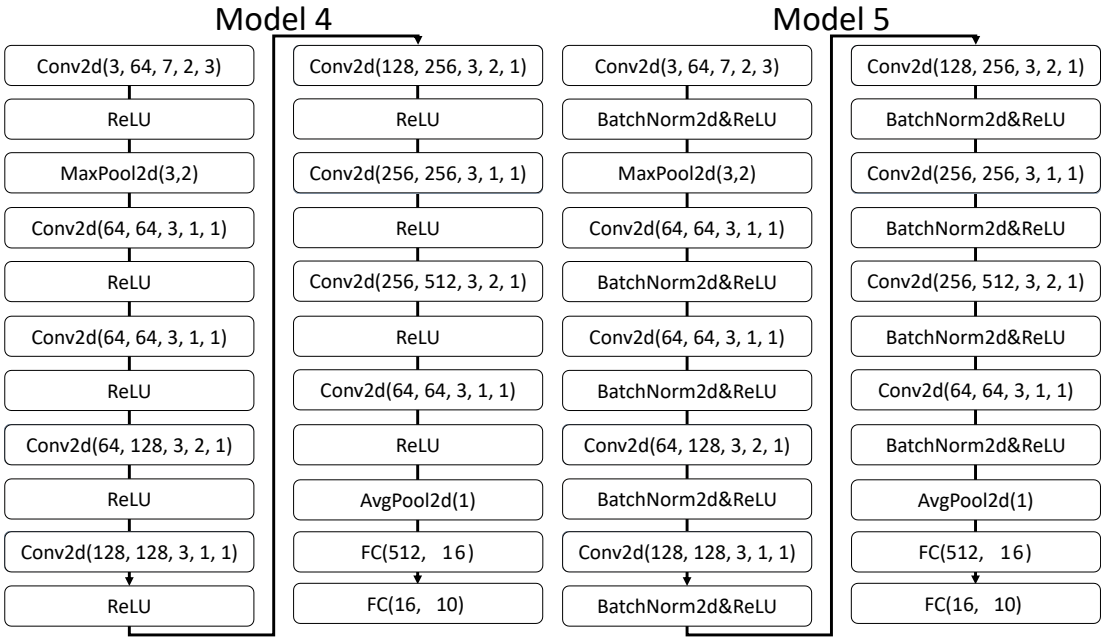

図 6.4　実験 2 で使用したニューラルネットワークモデル

デルは図 6.4 に示した Model 4 と Model 5 である．Batch Normalization 無しのニューラルネットワークモデルが Model 4，Batch Normalization 有りのニューラルネットワークモデルが Model 5 である．Model 5 は Batch Normalization を追加した Model 4 であるため，Model 4 の AUROC が高く，Model 5 の AUROC が低下することが確認できれば，AUROC 低下の原因は Batch Normalization であると断定できる．

Model 4 と Model 5 の AUROC の比較結果を表 6.3 に示す．表 6.3 は各データセット，各 σ の値における Batch Normalization を含むニューラルネットワークの AUROC(%) の値と，含まないニューラルネットワークの AUROC(%) の値を比較した表である．表 6.3 を見ると，Batch Normalization 無しの場合は 6.2.2 節で得られた AUROC の値と変わらない値が出ており，提案手法による外れ値検出ができているといえる．それに対して，Batch Normalization 有りの場合は 6.2.2 節で得られた AUROC の値よりも低い値が出ており，その AUROC の値はランダ

ム予測 (50%) と変らない．よって，提案手法による外れ値検出ができていないといえる．以上より，AUROC が低下する原因は Batch Normalization であるという我々の主張は正しいといえる．

6.3.4 問題点への対応策

前節にて我々は，AUROC が低下した原因は Batch Normalization であると判断した．より詳細な原因は，5.1 節で説明したデータセットの学習フェーズと，5.2 節で説明した一つの教師データを用いた追加学習フェーズでは，ミニバッチに含まれる教師データの分布に違いが生じていることである．我々は，使用するニューラルネットワークモデルによって，AUROC が低下する場合があることは問題であると考える．そこで，この問題点を克服できるように提案手法の実装を変更することにした．以下で考え方を説明し，実装の変更については 6.4.1 節で説明する．

上記の問題点が発生する原因はミニバッチに含まれる教師データの分布に差が生まれることである．この差を無くすためには，以下の二つの方法のうち，どちらかの案を適用する必要がある．一つ目は，5.1 節で説明したデータセットの学習フェーズを変更して，5.2 節で説明した一つの教師データを用いた追加学習フェーズのミニバッチに含まれる教師データの分布に近づけるような方法を考案することである．二つ目は，5.2 節で説明した一つの教師データを用いた追加学習フェーズを変更して，5.1 節で説明したデータセットの学習フェーズのミニバッチに含まれる教師データの分布に近づけるような方法を考案することである．

まず我々は，一つ目の方針でミニバッチに含まれる教師データの分布を近づける方法を考えた．その場合，データセットの学習方法を変更する必要がある．しかし，

表 6.3　Batch Normalization を含むニューラルネットワークモデルと，含まないニューラルネットワークモデルの AUROC(%) の比較

Dataset	SVHN		CIFAR10	
σ	0.0	0.4	0.0	0.4
Batch Normalization 無し	94.88	93.45	85.61	82.37
Batch Normalization 有り	52.84	53.73	49.74	48.50

5.1 節で説明したデータセットの学習方法は，一般的なニューラルネットワークモデルの構築方法であるため，変更することは難しいと考えた．そこで我々は，二つ目の方針でミニバッチに含まれる教師データの分布を近づける方法を考えた．我々のアイデアは，5.1 節で説明したデータセットの学習方法で作成されるミニバッチをベースにしつつ，そのミニバッチに D_{test} から取り出した教師データ D_p をミニバッチに含まれるの教師データと置換することである．実装の方法は 6.4.1 節で説明する．

我々のアイデアが問題点を解決すると考えた理由について説明する．問題点が発生する原因は，ミニバッチに含まれる教師データの分布に差が生まれることである．その差は教師ラベル，入力データが複数種類存在するかしないかという差である．5.1 節で説明したデータセットの学習方法で作成されるミニバッチに含まれるの教師データの一部と D_{test} から取り出した教師データ D_p を置換する上記のアイデアは，ミニバッチに含まれる教師ラベル，入力データが複数種類存在する状態を保持できる．そのため，ミニバッチに含まれる教師データの分布が大きく変わらず，Batch Normalization を使用する際の問題点が解消できると考えた．

6.4 実験 3：Batch Normalization への対応

本節では，6.3.4 節で説明した方法を基に，提案手法の実装を変更したときの外れ値検出の精度 (AUROC) を調べる実験を行う．この実験の目的は二つある．一つ目は，6.3.4 節で考案した対策を適用したとき，ニューラルネットワークモデルが Batch Normalization を含むか否かに関わらず提案手法を適用できるかどうかを調べることである．二つ目は，D_{test} から取り出した D_p に置換するミニバッチに含まれるの教師データ数（以下，置換数と呼称）を変化させたとき，提案手法の AUROC がどのように変化するかを調べ，適切な設定を特定することである．置換数についての詳細は 6.4.1 節で述べる．6.4.1 節では，6.3.4 節で説明した方法を適用した場合，5.2 節で説明した実装がどう変更されるのかについて説明する．6.4.2 節では，実験設定について説明する．6.4.3 節では，実験の結果を報告し，その結果に対する考察を 6.4.4 節で行う．

6.4.1　対応策適用後の特徴量抽出

本節では，5.2 節で説明した実装の変更点について述べる．5.2 節で説明した追加学習を行う方法を廃止し，新たな追加学習の方法を提案する．我々のアイデアは，5.1 節で説明したミニバッチ B のデータの一部を 5.2 節で説明した D_p に置換することである．以下で数式を用いた詳細な説明を行う．

5.2 節では，

$$D_p = \begin{bmatrix} i_p & s_p & t_p \end{bmatrix} \tag{6.4.1}$$

だけを用いて M_b の追加学習を行うと説明した．そして，その追加学習時のミニバッチの数式は，5.1 節で説明したミニバッチとは異なり，

$$B = \begin{bmatrix} i_p & s_p & t_p \\ \vdots & \vdots & \vdots \\ i_p & s_p & t_p \end{bmatrix} \tag{6.4.2}$$

となると説明した．

上記の方法による追加学習を廃止し，以下のように変更する．

$$D_p = \begin{bmatrix} i_p & s_p & t_p \end{bmatrix} \tag{6.4.3}$$

を用いて M_b の追加学習を行う．学習時のミニバッチの数式は 5.1 節で説明したミニバッチの一部を D_p に置換した

$$B = \begin{bmatrix} i_{b1} & s_{b1} & t_{b1} \\ \vdots & \vdots & \vdots \\ i_p & s_p & t_p \\ \vdots & \vdots & \vdots \\ i_{bm} & s_{bm} & t_{bm} \end{bmatrix} \tag{6.4.4}$$

を使用する．ただし，ミニバッチに含まれる教師データと，D_{test} から取り出した D_p を任意の個数置換することができ，置換を行う教師データはランダムとする．例えば，バッチサイズ 4 のミニバッチに対して，置換数を 2 に設定すると，ミニ

バッチは

$$B = \begin{bmatrix} i_{b1} & s_{b1} & t_{b1} \\ i_p & s_p & t_p \\ i_{b3} & s_{b3} & t_{b3} \\ i_p & s_p & t_p \end{bmatrix}, \begin{bmatrix} i_p & s_p & t_p \\ i_{b2} & s_{b2} & t_{b2} \\ i_p & s_p & t_p \\ i_{b4} & s_{b4} & t_{b4} \end{bmatrix}, \begin{bmatrix} i_{b1} & s_{b1} & t_{b1} \\ i_{b2} & s_{b2} & t_{b2} \\ i_p & s_p & t_p \\ i_p & s_p & t_p \end{bmatrix} \tag{6.4.5}$$

などで表されることになる．

以上に示した対応策によって Batch Normalization への対応を行う．5.2 節でも説明した通り，M_b が i_p を s_p であると判定できるようになるまで学習を繰り返し，学習終了後の M_b を M_p と表す．

6.4.2 実験設定

本節固有の実験設定（使用データセット，ダミーデータセット生成用パラメータ，使用ニューラルネットワークモデル，D_p の置換数）について述べる．本節固有ではない実験設定については 6.2.1 節を参照されたい．

使用したデータセットは SVHN，CIFAR10，CIFAR100 の 3 種類である．ダミーデータセット生成用パラメータはミスラベル率 0.2，スパース性を [0, 0.4] に設定した．使用したニューラルネットワークモデルは，図 6.1 の Model 2,Model 3 と，図 6.4 の Model 5，図 6.5 の Model 6 である．SVHN，CIFAR10 を使用する際に Model 2，Model 5 を使用し，CIFAR100 を使用する際に Model 3，Model 6 を使用した．これらのモデルを用意した理由は三つある．一つ目の理由は，Batch Normalization を含むニューラルネットワークモデルに適用可能であることを示すためである．そのため，Model 5，Model 6 を用意した．二つ目の理由は，Batch Normalization を含まないニューラルネットワークモデルでも適用可能な方法であることを示すためである．Batch Normalization によっておこる問題点への対応策として 6.4.1 節で説明した変更を行った．この変更によって Batch Normalization を含まないニューラルネットワークモデルに提案手法が適用できなくなる可能性は排除できていない．そのため，Batch Normalization を含まないニューラルネットワークモデルとして Model 2，Model 3 を用意した．三つ目の理由は，6.2 節で行った実験の結果と比較するためである．本節で行う Model2，Model3 を用いた実験と 6.2 節で行った実験は同じモデルを使用しており，前提条件も同じであるた

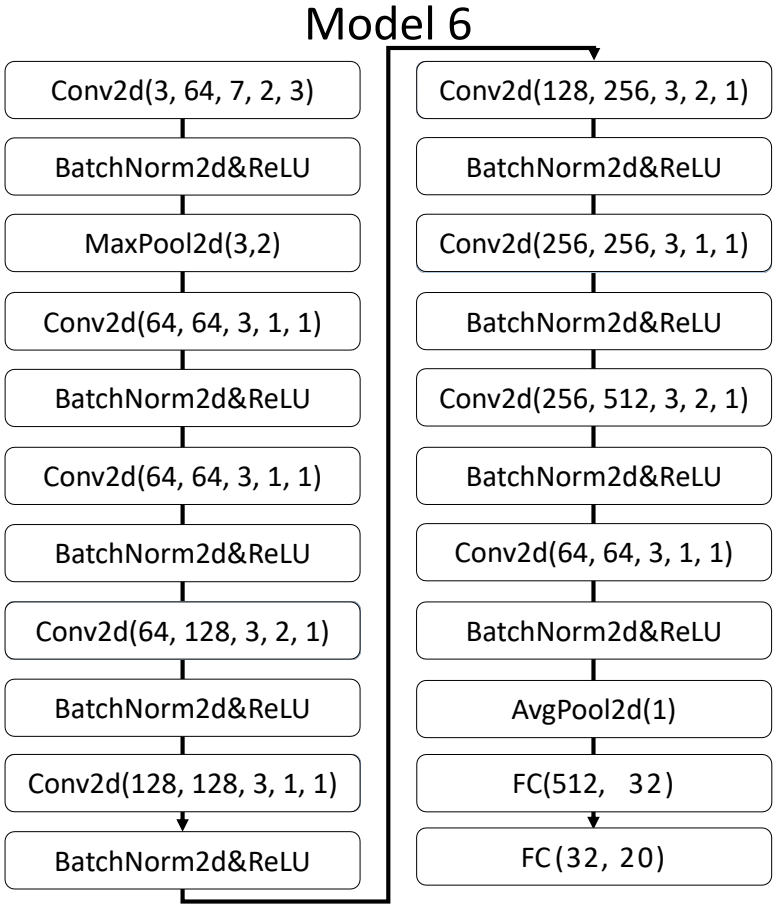

図 6.5　実験 3 で使用したニューラルネットワークモデル

め，結果を直接比較することができる．そのため，Batch Normalization に対応した提案手法であっても 6.2 節で比較した既存手法よりも優れていると主張することが可能である．D_p の置換数は [1,2,4,8,16] の設定で実験を行う．置換数の上限を 16 に設定した理由は以下の通りである．データセットの学習におけるミニバッチに含まれる教師データの分布と，追加学習におけるミニバッチに含まれる教師データの分布は差があると考えられる．そして，その差が生まれることによって AUROC が低下すると考えられる．そのため我々は，最低でも 50% はデータセットの学習フェーズのミニバッチと同じ教師データを使用するべきであると考えた．

6.4.3 結果

実験の結果を表 6.4 に示す．表 6.4 は各条件における AUROC(%) の値を示している．Batch Normalization の項目が無しの場合は Model 2 または Model 3 を使用しており，有りの場合は Model 5 または Model 6 を使用している．ρ，σ，Dataset はそれぞれミスラベル率，スパース性，使用したデータセットを示している．Replace = は置換数を示している．

得られた結果からわかる事実は以下の四つである．

- 提案手法に対応策を適用することによって，Batch Normalization 有りのモデルを使用した場合でも提案手法が適用可能になった．
- 対応策適用後の提案手法を Batch Normalization 無しのニューラルネットワークモデルに適用た場合，6.2 節の実験結果よりも AUORC が高い確認した．
- 表 6.1 と表 6.4 の Batch Normalization の項目が無しの場合を比較したとき，対応策を適用したほうが AUROC が高いことを確認した．
- 表 6.4 の Replace = 1 から Replace = 16 の結果を見ると，Replace の値が小さくなるほど AUROC が向上する傾向があることを確認した．

次節では，この結果を踏まえた考察を行う．

表 6.4　対応策適用後の外れ値検出の精度を調べる実験の結果 (AUROC(%))

Dataset	SVHN				CIFAR10				CIFAR100			
Batch Normalization	無し		有り		無し		有り		無し		有り	
σ	0.0	0.4	0.0	0.4	0.0	0.4	0.0	0.4	0.0	0.4	0.0	0.4
Replace = 1	96.94	96.25	97.89	97.16	88.13	85.53	92.54	90.67	84.15	82.21	88.35	85.95
Replace = 2	96.67	95.66	97.28	96.45	87.78	84.79	91.84	89.93	83.91	81.12	88.07	85.59
Replace = 4	95.62	94.20	96.87	96.06	86.82	83.39	91.54	89.59	82.89	79.52	87.89	85.39
Replace = 8	94.62	92.99	96.80	96.01	85.67	82.02	91.34	89.27	81.77	78.16	87.61	85.11
Replace = 16	94.02	92.35	96.84	95.98	84.91	81.18	91.18	88.97	80.94	77.28	86.72	84.46

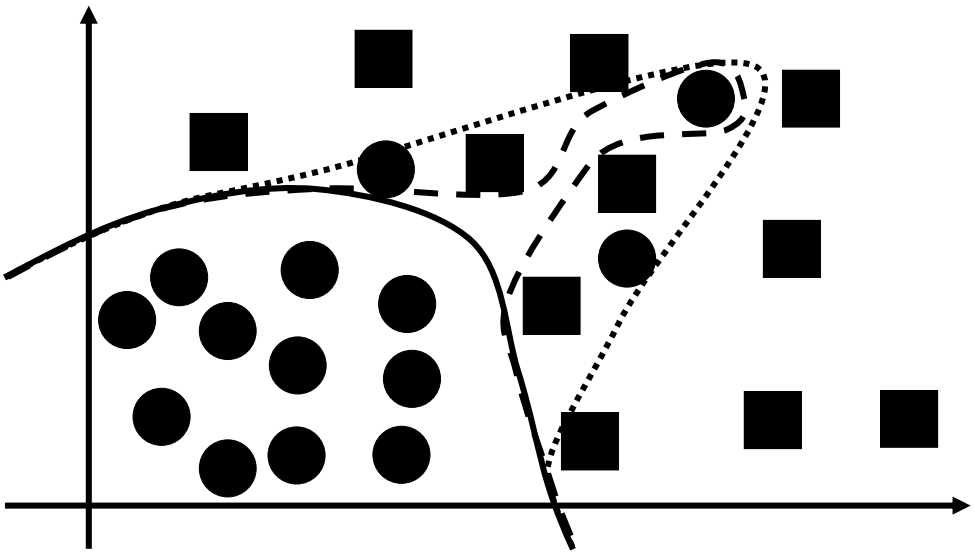

図 6.6　対応策適用前後の M_p の分類境界のイメージ図（対応策適用前：点線，対応策適用後：破線）

6.4.4　考察

6.4.3 節で述べた通り，提案手法に対応策を適用することによって，Batch Normalization 有りのモデルを使用した場合でも提案手法が適用可能になった．そして，Batch Normalization 無しのモデルでも対応策を適用した提案手法が適用可能であることを確認した．以上のことから，対応策を適用した提案手法は，どのようなニューラルネットワークモデルであっても適用可能であると考えられる．ただし，本稿で行った実験は画像だけを扱っているため，提案手法が適用可能なドメインは画像に限られることに注意されたい．適用範囲についての詳細は，7 章で述べる．

表 6.1（対応策適用前）と表 6.4 の Batch Normalization の項目が無しの場合（対応策適用後）を比較したとき，対応策適用後の方が AUROC が高いことを確認した．上記の結果から，対応策を適用すると，我々が対応策適用前の提案手法よりも優れた手法になると考えられる．対応策が優れている理由について我々の考えを図 6.6 に示したイメージ図を用いて説明する．M_b に追加学習を行う際に D_{test} の教師データのみを学習する (対応策適用前の提案手法) 場合，他の教師データを考慮せ

ずに分類境界を更新することになる．そのため，M_p の分類境界は，M_b の分類境界の外に存在する他の教師データも同時に含むように更新される可能性がある．つまり，M_p の分類境界は，学習した教師データを含む代わりに他の教師データを誤る分類境界になっていると考えられる．この分類境界が図 6.6 における点線の分類境界である．対して，M_b に追加学習を行う際に D_{train} のミニバッチに含まれる教師データの一部を D_{test} の教師データに置換して学習する (対応策適用後の提案手法) 場合，他の教師データを考慮しつつ，分類境界を更新することになる．そのため，M_p の分類境界は，M_b の分類境界の外に存在する他の教師データを含まないように更新される．つまり，M_p の分類境界は，学習した教師データを含み，他の教師データも誤らない分類境界になっていると考えられる．この分類境界が図 6.6 における破線の分類境界である．以上の考えを基に，我々は対応策適用前後の提案手法について以下のことが言えると考えた．対応策適用後の提案手法によって得られる特徴量は，対応策適用前の提案手法によって得られる特徴量と比較して，追加学習を行う教師データのみを含む分類境界を表す特徴量に近い．そのため，AUROC の向上という結果が得られた．

表 6.4 の Replace = 1 から 16 の結果を見ると，Replace の値が小さくなるほど AUROC が向上する傾向があることを確認した．この結果は，6.3.3 節で我々が述べた考えの裏付けとなる結果である．6.3.3 節にて我々は，データセットの学習フェーズにおけるミニバッチに含まれる教師データの分布と，追加学習におけるミニバッチに含まれる教師データの分布がずれることが AUROC 低下の原因であると主張した．追加学習を行う際，Replace の値を大きくする程，同じ教師データがミニバッチに含まれる個数は増加する．つまり，Replace の値を大きくする程ミニバッチに含まれる教師データの分布のずれが大きくなるといえる．そして，得られた結果は Replace の値を大きくする程 AUROC が低下するという結果であるため，我々の考えを支持する結果といえる．

表 6.4 の Replace = 1 から Replace = 16 の実験を行う中で，我々は Replace の値を 1 や 2 に設定すると特徴量抽出（追加学習）に時間が掛かるという問題点があることが分かった．そこで，Replace = 1 から Replace = 16 の条件において，追加学習の上限エポック数を変化させたとき，AUROC が頭打ちになる上限エポック数を調べた．その結果，表 6.4 に示したいかなる条件であっても，2 エポック以

上は上限エポック数を増やしても AUROC は変化しないことが分かった．上限エポック数の設定はデータセット依存である可能性が排除できないため断言はできないが，我々は追加学習の上限エポック数を 2 に設定すると，AUROC の向上が見込めない無駄な処理時間を省くことができる可能性があると考える．

第 7 章　提案手法の適用範囲

本章では，我々は提案手法の適用範囲について説明する．ここでの提案手法は対応策適用後の提案手法を指している．まず，機械学習モデルについて述べ，次にデータセットについて述べる．

提案手法を適用可能な機械学習モデルは，ニューラルネットワークモデルである．また，5.2 節で述べた通り，予備実験としてロジスティック回帰モデルに提案手法を適用可能であることを確認している．その他の機械学習モデルについて，提案手法を適用可能かどうか調べる実験を行っていないため，適用可能かどうかは不明である．我々は，ロジスティック回帰のように提案手法の考え方をそのまま利用できる手法であれば適用可能であると考えている．6.2 節で述べた通り，対応策適用前はニューラルネットワークを用いる場合，AUROC が低下するニューラルネットワークモデルが複数存在するという問題があった．そこで 6.3 節では我々は対応策を考案し，6.4 節では我々は対応策によって問題が解決されたことを示した．そのため，対応策適用後の提案手法は性能が低下すること無く，様々なニューラルネットワークモデルに利用可能であるといえる．また，本稿の実験で触れられていないニューラルネットワークの構造を含む場合，提案手法を適用できるかどうかは不明である．

提案手法を適用可能なデータセットは以下の三つの条件を満たしている必要がある．

- 入力データが画像のデータセットであること
- シングルラベルのデータセットであること
- 分類タスクを解くためのデータセットであること

上記の条件を満たさないデータセットについて，提案手法を適用可能かどうか調べる実験を行っていないため，適用可能かどうかは不明である．我々は，使用データセットの種類を増やして実験を行うことで適用範囲の特定が可能であると考える．

第 8 章　おわりに

本稿では，我々は公開されているデータセットの中に誤った教師データが存在することを問題視し，データクレンジングの精度向上を目的とした研究を行った．我々は，複数のクラス $a, b, \cdots$ へデータを分類する問題を考えたとき，あるクラス a の分類境界外に存在する教師データのうち教師ラベルがクラス a である教師データは，クラス a の分類境界から遠いほど誤った教師データである可能性が高いという仮説を立てた．そして我々は，学習済みモデルから得ることができる分類境界の特徴量を用いたデータクレンジング手法を提案した．1 章，4 章，5 章で述べた通り，提案手法のアイデアは，ニューラルネットワークモデルの分類境界の特徴量はパラメータから得られるという考えである．そして，ある一つの教師データを用いて学習済みモデルに追加学習を行うことによって，データクレンジングのための外れ値検出に使用する特徴量を考案した．

6.2 節で行った実験において我々は，提案手法と既存手法である Confident Learning[3]，Label Fix[4] の AUROC を比較し，提案手法の優位性を示した．6.3 節で行った実験において我々は，提案手法の問題点は Batch Normalization が原因であると特定した．そして我々は，ニューラルネットワークモデルに Batch Normalization が含まれていても問題が発生しないように提案手法を改善し，6.4 節で行った実験によって提案手法が改善されたことを示した．

今後の展望として，7 章で述べた提案手法の適用範囲を明確にする実験を行うべきであると考えている．現時点では，利用可能なモデルも利用可能なデータセットも適用可能かどうかが不明な部分がある．我々は，その穴を埋める作業が行われるべきであると考える．また，我々が今回提案したニューラルネットワークモデルのパラメータを特徴量とする手法はあまり見かけない手法である．そのため，我々はニューラルネットワークのパラメータという特徴量が別の場面で利用できないか考えたい．

謝辞

本研究を進めるにあたって，指導教員である，鈴木優准教授に様々なご指導，ご助言を賜りました．研究テーマの立案から，論文執筆まで，全てにおいて助言をいただきました．時には意見が食い違うこともありましたが，議論を交わし，結論を出す過程は結構楽しかったです．面談に行った時に雑談をしたり，一緒にご飯を食べたり，いろいろな思い出があります．学士 3 年後期から修士 2 年までの 3 年半の間，くそお世話になりました！就職してからも，暇な時は遊びにきます，絶対に．

事務補佐員の井尾さん，佐野さんには，外部での研究発表を行う際の様々な手続きをするにあたり，お世話になりました．お二人には，研究室で行う行事の準備もしていただきましたし，大学から雇用される際の手続きなどもしていただきました．私たち学生が日々困ることなく過ごすことができているのはお二人の存在が大きいと思います．井尾さんには，合計で 2 年半お世話になりました．井尾さんとはファイナルファンタジーの話をもう少ししたかったですね．FFX は昔からずっと好きなので，同志がいて嬉しかったです．佐野さんは井尾さんが休んでいた期間（1 年）お世話になりました．まだ原神やってますかね？私は最近やってません．就職してからお金貯めてパソコンを新調したら再開する予定です．

鈴木研究室に所属する皆さんには，普段のゼミで研究内容について意見をいただいたり，実験を行うにあたって用意するデータの作成を手伝っていただいたりしました．感謝いたします．

B3 の皆さんはまだ研究室に来て半年もたっていないですが，みんな頑張ってるなぁという印象です．私が B3 の頃に比べて研究テーマが決まるのが早かったように感じます．みんな優秀そうなので，これからの研究を頑張ってください．あと，研究のことでわからないことがあったら遠慮なく先輩に聞いてください．鈴木研は優秀な先輩ばかりなので，誰に聞いてもちゃんと教えてくれると思います．

B4 の皆さんは良くも悪くも個性豊かで面白い人が多いなという印象です．皆修士に進学するので，後 2 年は鈴木研に在籍することになると思います．いろいろ大変なことはあると思いますが，B4 は仲がいいので，みんなで助け合って頑張って乗り越えてください．あと城所君は，先生と変なところで衝突するのではなく，研究についての議論で衝突してほしいですね．

M1 の皆さんはみんな優秀という印象です．私よりもたくさんの知識を持っている北村君をはじめ，国際会議に通った桑原君と太田さん（桑原君は論文誌も），海外留学するエルゲン君 · · · やっぱりみんな優秀ですね．これから後 1 年，就活も大変だと思いますが，頑張って乗り越えてください！

M1 の中でも特に，北村君と桑原君は私の研究の実験，実験結果の考察で相談に乗ってもらいました．私の主張は変じゃないか，この主張をするためにはどんな実験をしたらいいか，結果がこうなった原因は · · · といろいろなことを考える上で助けてもらいました．特に実験 3 は共著：北村みたいなものです．私にとっては第二の先生のような存在です．ありがとうございました．

同期の M2 の二人はおもろいやつとツイ廃（X 廃？）という印象です．沢田君も小林君も編入という形で M1 からの同期になったけど，わりとすぐに仲良くなった気がします．3 人ともよく喋るからかな？沢田君は鈴木研に来る前は機械学習系の研究をやっていなかったのにも関わらず，知識も身につけて，着実に研究を進めてここまでやってきたということがすごいなと思っています．純粋に尊敬してます．小林君は，うーん，ほどほどに頑張ってたけど，もうちょっと努力の方向が良い方向に向いてたら良かったのかなと思います．これからはみんな別々の会社に入り，社会人として生きていくことになります．今までのように毎日会うことはなくなりますが，定期的に会いましょうね！

最後に、これまで経済的・心身的に支えて下さった家族に深く感謝し，お礼を申し上げます．

本論文を書き終えることができたのは，皆様が支えてくださったおかげです．心より感謝申し上げます．

参考文献

[1] C. Müller Andreas, Guido Sarah, 訳：中田秀基. Python で始める機械学習-scikit-learn で学ぶ特徴量エンジニアリングと機械学習の基礎（原タイトル: Introduction to Machine Learning with Python A Guide for Data Scientists）. オライリー・ジャパン（元出版：O'Reilly Media, Inc.）, 2017.

[2] Curtis Northcutt, et al. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, Vol. 1, pp. 1–24, 2021.

[3] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, Vol. 70, pp. 1373–1411, 2021.

[4] Nicolas M. Müller and Karla Markert. Identifying mislabeled instances in classification datasets. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2019.

[5] 和田かず美. 多変量外れ値の検出–msd 法とその改良手法について. 統計研究彙報, Vol. 67, pp. 89–157, 03 2010.

[6] Kaiming He, et al. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[7] Jacob Devlin, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc.NAACL-HLT, Volume 1*, pp. 4171–4186, 2019.

[8] Lorenzo Bruzzone and Claudio Persello. A novel context-sensitive semisupervised svm classifier robust to mislabeled training samples. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 47, No. 7, pp. 2142–2154, 2009.

[9] Ishan Jindal, Daniel Pressel, Brian Lester, and Matthew Nokleby. An effective label noise model for DNN text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 3246–3256, Minneapolis, Minnesota, June 2019.

[10] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin Mcguinness. Unsupervised label noise modeling and loss correction. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97 of *Proceedings of Machine Learning Research*, pp. 312–321. PMLR, 09–15 Jun 2019.

[11] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2021.

[12] Jaree Thongkam, et al. Support vector machine for outlier detection in breast cancer survivability prediction. In *Advanced Web and Network Technologies, and Applications*, pp. 99–109, Berlin, Heidelberg, 2008.

[13] Sarah Jane Delany and Pádraig Cunningham. An analysis of case-base editing in a spam filtering system. In *Advances in Case-Based Reasoning*, pp. 128–141, 2004.

[14] Karishma Sharma, Pinar Donmez, Enming Luo, Yan Liu, and I. Zeki Yalniz. Noiserank: Unsupervised label noise reduction with dependence models. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pp. 737–753, Cham, 2020.

[15] Vaibhav Pulastya, Gaurav Nuti, Yash Kumar Atri, and Tanmoy Chakraborty. Assessing the quality of the datasets by identifying mislabeled samples. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '21, p. 18–22, New York, NY, USA, 2022.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, Vol. 20, No. 3, p. 273–297, 1995.

[17] Bernhard Schölkopf, et al. A generalized representer theorem. In *In*

Proceedings of the Annual Conference on Computational Learning Theory(COLT '01/EuroCOLT '01), pp. 416–426, 2001.

[18] David M. J. Tax and Robert P. W. Duin. Support vector data description. *Machine Learning*, Vol. 54, pp. 45–66, 2004.

[19] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI / PAMI)*, Vol. 20, No. 8, pp. 832–844, 1998.

[20] Heiko Hoffmann. Kernel pca for novelty detection. *Pattern Recognition*, Vol. 40, No. 3, pp. 863–874, 2007.

[21] Martin Ester, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining(KDD)*, p. 226–231, 1996.

[22] Zengyou He, et al. Discovering cluster-based local outliers. *Pattern Recognition Letters*, Vol. 24, No. 9, pp. 1641–1650, 2003.

[23] Siqi Wang, et al. Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32, pp. 1–14, 2019.

[24] Alex Krizhevsky, et al. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 25, pp. 1–9, 2012.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of The International Conference on Learning Representations (ICLR)*, pp. 1–14, 2015.

[26] G. Huang, et al. Densely connected convolutional networks. In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

[27] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Vol. 97, pp. 6105–6114, 2019.

発表リスト

[1] 三島惇也，鈴木優『外れ値検出手法を用いたクラウドワーカの品質推定手法の提案』，東海関西データベースワークショップ，2022
[2] 三島惇也，鈴木優『クラウドソーシングにおける作業結果のクレンジングによるデータセットの品質向上』，第 15 回データ工学と情報マネジメントに関するフォーラム，2023
[3] 三島惇也，諸橋玄武，深見匠，張一凡『自然言語モデルの脆弱性の検証と評価指標の提案』，第 27 回日本医療情報学会春季学術大会，2023
[4] 三島惇也，鈴木優『マルチラベル学習によるラベル付け時のばらつきを考慮した学習方法の提案』，東海関西データベースワークショップ，2023