機械学習品質評価・向上技術に関する報告書

機械学習品質マネジメントガイドライン付属文書

第 2 版

2022 年 8 月 2 日版

国立研究開発法人産業技術総合研究所

デジタルアーキテクチャ研究センター
テクニカルレポート DigiARC-TR-2022-06

サイバーフィジカルセキュリティ研究センター
テクニカルレポート CPSEC-TR-2022007

人工知能研究センター
テクニカルレポート

まえがき

国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）受託業務（JPNP 20006）「機械学習システムの品質評価指標・測定テストベッドの研究開発」では、機械学習品質を説明可能にするために機械学習品質マネジメントガイドライン[1]を開発しており、このガイドライン開発と並行して、機械学習品質の評価・向上技術の調査・研究開発も行っている。本調査・研究開発は現在も継続中であるが、機械学習品質マネジメントガイドラインに記載されている品質評価に関する技術的な知見も得られているため、本稿では、これまで（2019～2021 年度）の本調査・研究開発の内容と結果について報告する。

目次

1 はじめに

統計的機械学習を利用した各種産業製品の品質を明確に説明可能にするために、機械学習品質マネジメントガイドラインが開発されている[1]。このガイドライン第 3 版では、機械学習システムに対する 9 つの内部品質特性（学習モデルの安定性やプログラムの信頼性等）に着目しているが、これら内部品質特性の評価や向上の技術は必ずしもまだ確立していない。本稿は、ガイドラインの開発と並行して行われている内部品質特性の評価・向上技術の調査・研究開発に関する報告書である。

1.1 本調査・研究開発の概要

図 1.1 に、2019~2021 年度に調査・研究開発した機械学習品質評価・向上の技術（図 1.1 の中央の黄色い四角が本稿で説明する技術を表す）と機械学習モデルのライフサイクルの各フェーズ、9 つの内部品質特性との関係を示す。以下、各技術については 2 章以降で詳しく説明するが、ここで簡単に説明しておく。

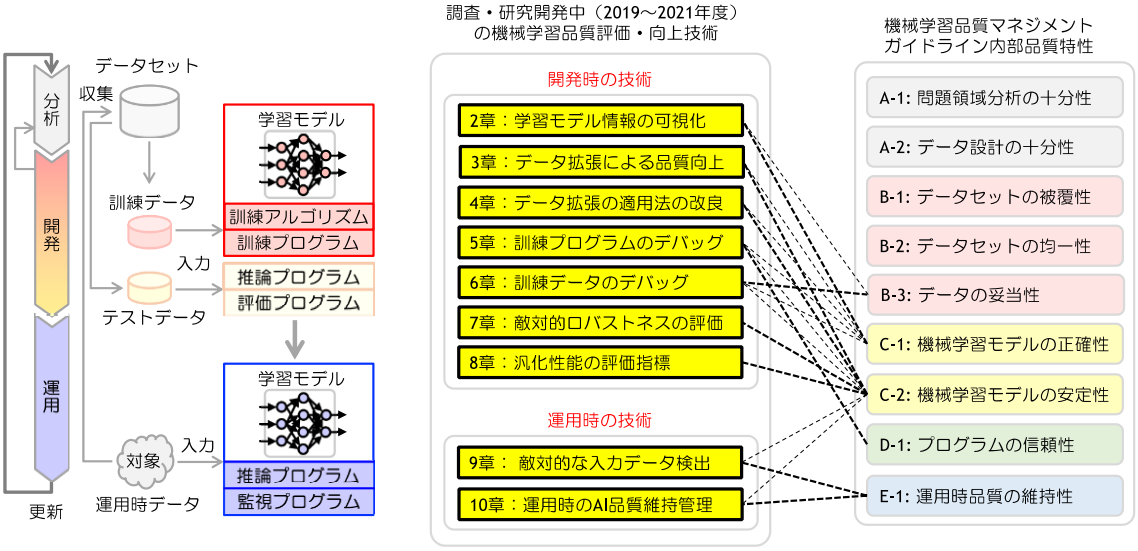


図 1.1 機械学習品質評価・向上技術（2019～2021 年度調査・研究開発）

- **2 章 学習モデル情報の可視化：**
  機械学習モデルの品質評価作業を支援するために、複数の学習モデル間の差分・比較結果の可視化や各モデルに反映されている作業者（アノテータ、モデル設計者）の感性の可視化を主な目的として、モデルの構造や精度に加え、モデル作成に関わる作業者の作業手順やモデルへの影響を複合的に可視化するツールの実装を進めた[2][3]。

- **3 章 データ拡張による品質向上：**
  学習データを加工してデータを増強するデータ拡張手法が画像識別の品質評価に与える影響とその品質改善方法について検討を行った。様々なデータ拡張手法（とその

組み合わせ）に対して実験を行い、識別精度の平均値だけでなく分散も計測することが、品質を評価するために重要であるとの結論を得た。

- **4 章　深層学習におけるデータ拡張の適用法の改良による品質改善：**
  データ拡張によって得られる多様性をさらに向上させ、深層学習モデルの正確性や安定性を向上させるための訓練手法として、FC-mixup 法や Latent DA 法等を提案し、画像識別問題においてネットワークモデルの品質向上に寄与することを確認した[4][5]。また、Latent DA 法については、データ拡張に最適な層を動的に発見する手法 AdaLASE を開発し、その効果を検証した。

- **5 章　深層 NN ソフトウェアのデバッグ・テスティング：**
  深層学習型の機械学習モデルの不具合の原因を推論時の直接原因（予測・推論プログラムによる）と訓練時の根本原因（訓練・学習プログラム、学習モデル、学習データセットによる）に分けて整理し、訓練・学習プログラムのバグの有無を、推論時の機械学習モデルの内部情報（ニューロンカバレッジ）によって推定するための検査指標と分析手法を提案し、実験によってその検査指標の有効性を確認した[6]-[10]。

- **6 章　訓練データのデバッグ・テスティング：**
  深層学習型の機械学習モデルの不具合の根本原因が訓練データの偏りにある場合を対象とし、訓練データの偏りをモデル正確性とモデル・ロバスト性の 2 つの品質観点から判断する方法を検討した。訓練データの偏りは、モデル内部の活性状態からニューロン・カバレッジの偏りから推定する方法を提案し、実験によって訓練データのデバッグに役立つ情報が得られることを確認した。

- **7 章　ロバストネスの評価・向上技術：**
  敵対的データを含む入力ノイズに対するロバストネスの指標の一つに最大安全半径（誤判断を生じないことを保証できるノイズの最大値）を計測する技術と、その半径を増加させる技術について調査した。最大安全半径の評価の計算コストは非常に高いが、近年、その近似値を効率よく見積もる手法が提案されてきている。

- **8 章　汎化誤差上界の見積り技術：**
  機械学習モデルの汎化性能を評価するため、全入力に対する不正解率（汎化誤差）の上界の見積り方法を調査した。従来の汎化誤差上界の見積り方法の精度は低く、汎化性能の評価指標として適用することは本報告書執筆時点（2021 年度）ではまだ難しいが、最近、見積り精度は改善されてきており、将来的には汎化性能評価技術として期待できる。

- **9 章　敵対的データ検出技術：**
  運用時に入力画像が敵対的データであるかを判別する方法を実用的に確立するために、最先端の敵対的データ検出手法を調査し、4 つの主要なカテゴリに分類するとともに追実験を行い、NIC 法が最も高い検出率を示すことを確認した。そこで、NIC 法

に基づく敵対的データ検出を行うために NIC フレームワークを構築し、その高い検出率の理由を説明するために Kullback-Leibler 情報量を用いて NIC 法を評価した。

- **10 章　運用時における AI 品質管理技術：**
  運用時に発生するデータの変化や未知のデータの到来に対応するために、データ分布変化に対する検知・適応技術および同技術の関連技術であるドメイン適応技術の最新技術を幅広く調査した。適用可能性や運用コストの面から有望な教師なしコンセプトドリフト適応技術、訓練データに依存しない適応技術やラベルシフトなどの入力データの分布以外の変化にも対応可能な教師なしドメイン適応技術の研究開発が行われているおり、この調査結果をサーベイ[11][12]としてまとめた。

1.2　著者リスト

各章の著者は以下のとおり：

- 1 章：磯部 祥尚　（産業技術総合研究所）
- 2 章：宮城 優里　（産業技術総合研究所）
- 3 章：大西 正輝　（産業技術総合研究所）
- 4 章：高瀬 朝海　（産業技術総合研究所）
- 5 章：中島 震　　（国立情報学研究所）
- 6 章：中島 震　　（国立情報学研究所）
- 7 章：磯部 祥尚　（産業技術総合研究所）
- 8 章：磯部 祥尚　（産業技術総合研究所）
- 9 章：中島 裕生、西田 啓一　（テクマトリックス株式会社）
- 10 章：大川 佳寛、小林 健一　（富士通株式会社）

1.3　謝辞

本調査・研究開発は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）受託業務（JPNP20006）「機械学習システムの品質評価指標・測定テストベッドの研究開発」として行っています。

2 機械学習モデル情報の可視化

ブラックボックスである機械学習モデルの構造や挙動についての分析を支援する手法として、情報可視化技術の導入が進んでいる。このような機械学習モデル可視化に関し、以下の 2 点を目的として新たな手法の研究を進めた。

- 複数のモデル間の差分・比較結果の可視化
  （人間が解釈・理解しやすい表現を用いた可視化の実装）
- モデルに反映されている作業者（アノテータ、モデル設計者）の感性の可視化
  （品質評価に用いることができる新たな要素の提案）

本章では、まず近年の機械学習モデル可視化技術の調査結果について解説する。続いて 2020～2021 年度に試作した、モデルと作業者情報を観察するための可視化ツールの実行例と今後の実装方針について報告する。

2.1 機械学習支援のための可視化手法についての調査

機械学習を対象とした可視化手法の基本的な目的はモデルの解釈可能性の向上であり、近年注目されている XAI（Explainable AI）と大きな関連がある。XAI の定義や評価方法について確定的なものは未だ存在しない。しかし、XAI の分類を試みる論文等は多数発表されているため、これらに沿って可視化の目的や手法を考案することができる。[13]では解釈可能性を上げるためのアプローチについて以下の 4 通りに分類している。可視化が特に有効である項目は(2)(4)であり、研究事例も多いとみられる。

(1) 大局的な説明（簡易なモデルによる複雑なモデル構造の近似）
(2) 局所的な説明（モデルの出力結果に関する判断根拠の説明）
(3) 説明可能なモデルの設計（設計段階での可読性の高いモデルの作成）
(4) 深層学習モデルの説明（画像データ内でモデルが認識している部分のハイライトなど）

このような機械学習の可視化手法は継続的に研究されており、用途や対象事例が多様であるためサーベイ論文も増加している。例えば[14]では深層学習可視化手法について、5W1H の要素に沿う形で解説・分類している。深層学習可視化分野の全体的な方向性や課題についても多数提示しており、特に品質評価のための可視化手法の開発を目指す本研究と関わりの深いものとして「モデル評価のためのインタラクションの改良」「モデルに対する人の積極的な関与による解釈可能性の向上」などが挙げられる。

機械学習可視化の研究が進み、実社会でも利用される機会が増加するにつれて、1 つの可視化画面で複合的な分析が行われる傾向が強まっている。従来は単体のモデルの詳細な分析や、データ（[15]）かモデル構造（[16]）のいずれかに特化した可視化などが多かった。しかし近年はデータとモデル構造の複合的な可視化手法や、複数のモデルを比較することを目的とした可視化手法についても研究が進んでいる。機械学習モデルを構成する要素は膨大なものとなるため、例えばモデル単体の可視化結果をモデル個数分作成し、並べて比較するには大きな手間がかかる。また、比較しようとするモデル間の構造や精度の差が小さく、モデルの特徴などを見落とす可能性も考えられる。そのため、差分を強調する表現を用い、

限られた画面内で効率的に差分を発見できるようにする可視化手法の必要性が高い。(例えば[17]では、10 種類以上のモデルについてデータの入力から出力までのパイプライン、ハイパパラメータの値などを一画面で比較できる。)

ここまでで、モデル自体の性質や精度に関する可視化手法についての傾向と事例を紹介した。これと並行して、モデル作成に関わる作業者(アノテータ、設計者)とモデルとの関わり方についても調査を進めた。画像認識などの分野では人の認識能力を超える精度を持つモデルが開発される一方で、「モデルの精度を向上させるためには積極的な人の介入が望ましい」という提言は分野を問わず根強い。AI と人との関係性や、モデル作成過程での効果的な介入方法に関し、以下のような項目に注目して議論した論文が多く存在する。

- 学習過程での、モデルの精度を向上させるための操作(調整や評価)の導入方法
- 操作性が良く、作業者のモチベーションを維持しやすいインタフェースの設計方法
- 関連分野(認知科学、心理学など)との連携

一例として[18]では、モデルの評価・改善のためのフィードバックを課された作業者の心理状態について検証しており、共通する傾向として「モデルに正しい処理手順を直接指示できることを好む」「自身の行動によってモデルの精度が改善されていることが可視化されると、モチベーションの向上やより積極的なフィードバックを見込める」などを紹介している。このような作業者自身の情報や、各作業者がモデルに与えた影響についての可視化事例は少ないという印象だが、以下のように品質保証の根拠として採用できると考えられる。

- 分析対象の事例、あるいは機械学習の専門家がモデル作成に参加していた場合、彼らの知識が十分にモデルの挙動に反映されていることを示す
- どの作業者の行動がモデルに強く反映されているのかを示し、調整するべき要素(データ、パラメータなど)を特定するための手がかりとする

2.2 機械学習モデルと作業者情報の可視化手法の提案

以上のような調査結果から、本研究では機械学習モデル可視化手法のうち特に「複数モデルの比較可視化」「作業者の感性に関する可視化」を重要視し、これらの性質を併せ持つような可視化ツールの設計を進めた。図 2.1 は提案手法の概要図である。

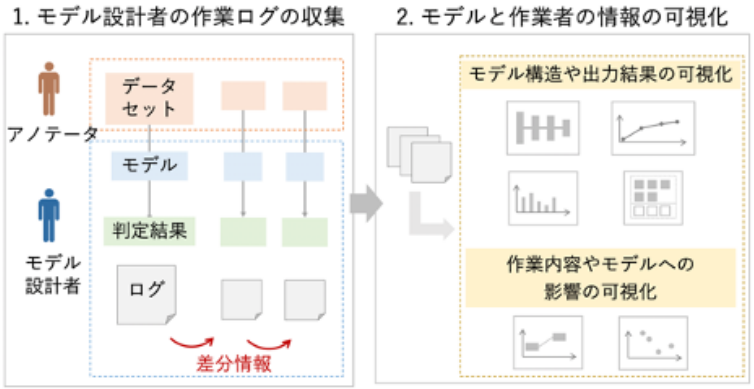


図 2.1 機械学習モデルと作業者情報の可視化手法の概要

2.2.1 モデル同士の差分に関するログの収集

まず、可視化の対象とするモデルの構造や調整過程、テスト結果などのログを収集する。現在の実装では事例として画像分類を想定し、機械学習の実験管理ツールである Cometl.ml を用いてモデル設計者のパラメータ調整過程やテスト結果などをテキストファイルとして取得している。アノテータについては直接的に作業ログを収集せず、モデル設計者がどのようにデータを選択し、前処理を適用したのかによって間接的に作業内容を評価する。

これらのログからモデル同士の差分（直前に使用したモデルからの変化量）を算出する。モデルに関する差分は「学習データ」「モデル構造」「最適化アルゴリズム」の 3 種類に分類しそれぞれ算出する。学習データの差分は、使用したデータやクラスの数、前処理に用いたパラメータの差分などを足し合わせて求める。モデル構造の差分は、2 つのモデルを構成するレイヤー同士のペアを作成し、各ペアの非類似度（レイヤーの種類やパラメータの差分）を合計することで求める。最適化アルゴリズムの差分については、アルゴリズムの種類が異なる場合には定数を付与し、同一である場合はパラメータの差分から計算する。このように 3 種類の差分を求めたのち、これらの値を合計したモデルの総合的な変化量も求める。

2.2.2 モデルの構造や作業者の情報の可視化

収集したログを用いて、モデルの構造や作業者の情報の可視化を行なう。本ツールの主なユーザはモデルの設計者と想定している。可視化に不慣れなユーザが含まれる可能性を考慮して基本的な可視化手法（折れ線グラフ、棒グラフなど）を組み合わせ、これらの連携（関連部分のハイライトなど）を積極的に行うという方針で実装を進めた。2020 年度にはモデルの構造や出力結果などの基本的な情報に関するビューを実装し、2021 年度には作業者によるモデルの調整やテストの経過を可視化するビューを作成した。

図 2.2 は 2020 年度に試作した可視化ビューの一覧であり、2 つの簡易なモデルについて MNIST の学習・出力結果を可視化した例である。

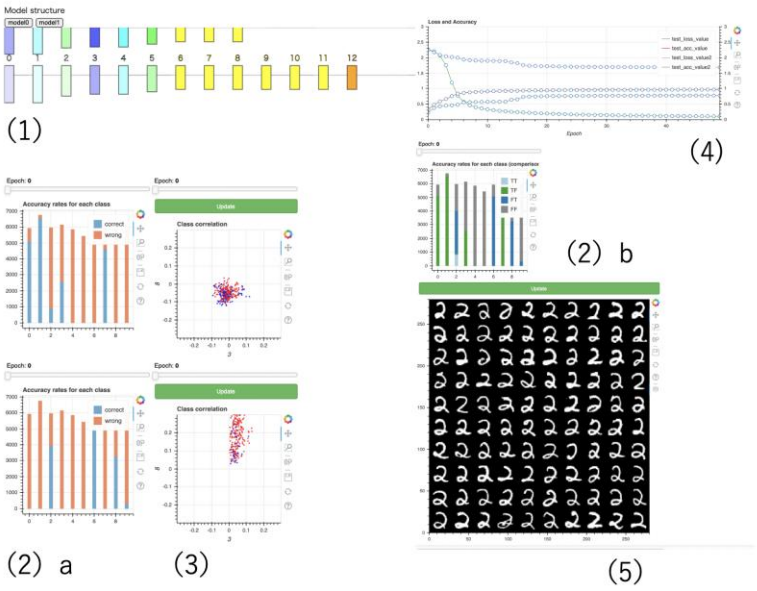


図 2.2 モデルの構造と出力結果に関する可視化ビューの一覧

JupyterLab 上で機械学習ライブラリ PyTorch、可視化ライブラリ Bokeh などを用いて以下のようにビューを作成し、2 つのモデルの特徴を比較できるようにした。

(1) 各モデル構造のネットワーク
(2) クラスごとの出力結果の棒グラフ
    a) モデルごとの可視化
    b) 2 つのモデル間の差分の可視化
(3) モデルごとの、選択した 2 クラス間の出力結果相関の散布図
(4) 精度の折れ線グラフ
(5) 特に高い（低い）確信度で分類されたデータのサムネイル一覧

図 2.3 は、2 つのモデルについて MNIST に対する出力を分類した結果の例である。横軸は 0〜9 のクラス、縦軸はデータ数を表す。それぞれの棒の色分けは、2 つのモデルについての正解（T）・不正解（F）の組み合わせを表したものである。TF（FT）は「モデル 1（2）のみ正しく分類できたデータ」を意味する。学習開始直後（図 2.3 左）は、モデル 1 がクラス 0、1、7、モデル 2 がクラス 2、6、8 での正解率が高く、それぞれのモデルが異なる強みを持っていたことがわかる。学習が進んだ段階（図 2.3 右）ではどちらのモデルも多くのクラスで正解率が高くなっている。特にモデル 1 はモデル 2 が不得意とするクラス 3、4、5、7 を含めて正答率が高く、この段階ではモデル 1 の学習の方がモデル 2 よりも進んでいることがわかる。

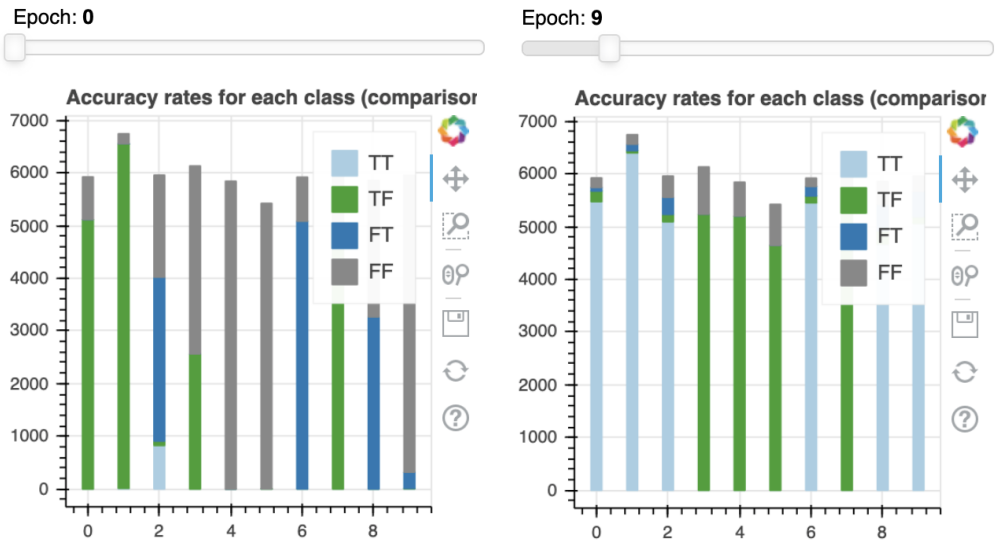


図 2.3　2 つのモデルの出力結果比較の例

続いて 2021 年度に作成した、モデルのテスト結果の時系列可視化機能を用いて生成した可視化結果を紹介する。図 2.4 は複数の画像分類モデルを順番にテストした際の、精度とモデル構造の変化量を可視化した図である。横軸はテストの順序を表し、縦軸は各モデルの top-1 を意味する。灰色〜黄色で着色したボックスが 1 つのモデルに対応している。最初に用いたモデルのボックスを除き、各ボックスの内部には 3 つの小さなアイコンを表示して

いる。これらのアイコンとボックスの色は、直前に使用したモデルからの変化量を表しており、彩度が高いほど変化が大きいことを意味する。具体的には、上段のアイコン（赤）が学習データ、中段のアイコン（青）がモデル構造、下段のアイコン（緑）が最適化アルゴリズムに関する変化量を表す。ボックス（黄）はこれらの変化量を総合した値を反映している。また各ボックスの中心座標をエッジで接続することで、精度の変化を明示する。

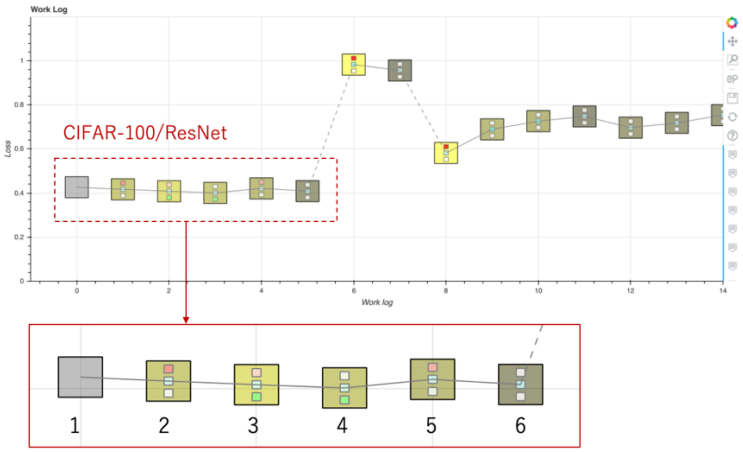


図 2.4　作業者のモデル調整履歴可視化の一例

表 2.1　作業履歴の可視化に用いたモデルのパラメータ設定

Index	l	m	p	a	d
1	0.1005	0.9	0	0.4	18
2	0.1005	0.9	0.45	0.4	18
3	0.06	0.5495	0.409	0.55	18
4	0.06	0.919	0.409	0.55	18
5	0.0335	0.919	0.2955	0.287	18
6	0.0335	0.919	0.2955	0.287	34

図 2.4 の赤枠で囲んだ部分は CIFAR-100 を学習した ResNet モデルを調整した過程の可視化結果である。[19]で実施されたハイパパラメータ調整のシナリオを参照し、表 2.1 のようにパラメータを変更しながらモデルの学習とテストを 6 回実行しログを収集した。lは学習率、mはモーメンタムの値である。p、aは学習データに Random Erasing を適用した際の Erasing probability と Max erasing area である。dは使用した ResNet モデルの深さである。

図 2.4 の赤枠内に記載した数字は表 2.1 の Index に対応する。2 と 5 では 3 つのアイコンのうち、学習データの変化を表す一番上のアイコンがオレンジ色でハイライトされている。モデル 1 から 2、および 4 から 5 に進む際に、学習データに関するパラメータであるp、aの値を大きく変更したことを反映している。 同様に、3 と 4 では一番下のアイコンが黄緑色にハイライトされており、最適化アルゴリズムに関する変化を示している。具体的には、l と mの変更を反映したものとなっている。またボックスの色を見ると、直前に使用したモデルからの変更量が最も大きかったのは 3 番目のモデルであることがわかる。モデル 2 と比較するとl、m、p、aの 4 つのパラメータが変更されている。一方でモデルの精度

については全体的に変化が少なく、今回のパラメータ調整手順の影響が限られていたことがわかる。

赤枠よりも右に位置するボックスは、学習データを MNIST、ImageNet に変更しテストを進めた際の経過をプロットしたものである。図 2.4 の中央付近に 2 つのボックスが明るい黄色でプロットされているが、これは使用するデータセットとモデルを変更したタイミングと一致している。このように、特定のモデルの細かな調整過程を示すのに加えて、複数の事例を用いた長期的な作業履歴を可視化することもできる。

2.3 今後の方針

今後は以下のような方針で可視化機能の拡張に取り組みたい。まず単独、または少数の機械学習モデル作成者（作業者）を対象として、モデルや作業内容の長期的な評価結果と改善策の推薦結果を可視化することを目指す。その後、多数の作業者の作業パターンの比較を行い、モデルまたは作業者同士の類似度や分類結果を俯瞰的に可視化する。これらのような可視化結果を観察することで、作業者のスキルレベルの推定や作業の特徴（作業パターン）の分類を行えるようにしたい。

また、現時点では手動でのモデル設計を前提としているが、NAS（Neural architecture search）などと組み合わせることで、より幅広い事例に本手法を適用できるように拡張することも有用と考えられる。

3 データ拡張による品質向上

深層学習を用いた画像認識の中でも物体の識別の品質評価に焦点をあてて研究を進める。学習データを加工することでデータを増強する Data Augmentation（データ拡張）が画像識別の品質評価にどのように影響を与えるのか、品質を改善するためにはどうすればいいのかについての検討を行った。

特に議論の中で「品質評価に関して論文では state-of-the-art を競うことが多いが、産業応用を考えた場合には精度が 98%という場合にはデータセットを変えた場合にでも 98%の精度が出ることが望ましい。それが保証されるなら 98%ではなく 80%でも構わない」という意見があった。これまでに研究会や国際会議で発表されている手法においてもデータセットに過学習することで精度が上がっているように見えるという問題は散見している。

3.1 学習データ数と識別率の関係（予備実験）

最初に予備実験として以下の 2 点を検証した。

予備実験① 学習データを増加していくことで識別率はどのように変化していくのか
予備実験② 学習データを変えることで識別率はどのように変化するのか

両予備実験において深層学習のモデルとして WideResNet28-10 を用い、データセットは CIFAR10 を用いる。CIFAR10 は 10 クラスの画像で構成され、各クラス毎に 6000 枚のデータ（合計 6 万枚）が用意されている。これらの画像を左右反転に加え、上下左右にずらすことでデータを 10 倍に増やし 60 万枚とし、この 60 万枚の世界がデータによって構成される全ての世界であると仮定する。予備実験①では 60 万枚の全ての世界の中から N 枚が観測されたとして学習を行い、全ての世界の 60 万枚で評価を行い識別率を求める。つまり N = 600000 で学習する際には世界の全てが観測できたと仮定して識別機を作成し、識別率を求めることとなる。これは十分に識別性能が発揮されるネットワークモデルにおいては識別率が 100% になることが報告されているが、実際にそうなるかを確かめる実験でもある。識別結果を図 3.1 に示す。横軸はデータ数の対数グラフ、縦軸は識別率である。

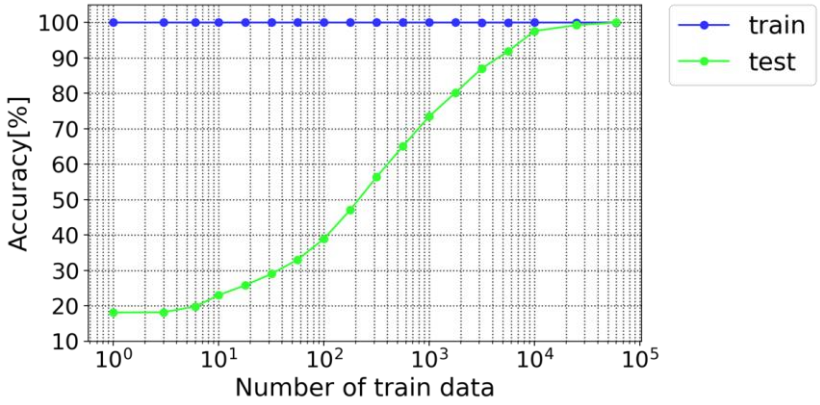


図 3.1 学習データ数と識別率の関係

これまでの研究で報告されているように 60 万画像という大量のデータであっても世界の全ての画像を学習データに利用することが可能であれば識別率は 100%を実現できることが分かる。また、各クラス 100 枚の学習データで 60 万枚の画像を識別すると 40%程度の識別精度であることが分かる。そこで予備実験②を行うための学習データ数は 10 クラス×100 枚の N = 1000 枚とした。オリジナルの各クラス 6000 枚の画像を 100 枚×60 セットに分割し、60 セットの学習データを用いてニューラルネットワークを学習し、60 万枚の画像で評価を行った。それぞれの学習データに対する識別精度のヒストグラムを図 3.2 に示す。

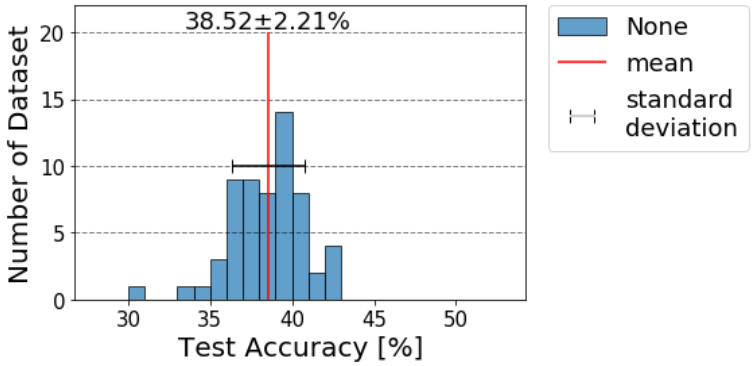


図 3.2 学習データの違いによる識別精度のヒストグラム

学習データがある組み合わせの場合には 42%の識別率が得られる一方で、学習データがある組み合わせの場合には識別率は 30%しか得られないことが分かり、これまでの識別精度だけの評価指標では品質の評価という観点では適していないことが明らかになった。なお、平均は 38.52%であり標準偏差は 2.21 である。この予備実験の結果から品質を評価する際には従来のような識別率だけで評価するのではなく、品質の安定性を評価するためには分散や標準偏差を考慮した評価が必要であることが分かる。

3.2 識別率の平均値と標準偏差の評価

これまでに提案されている様々なデータ拡張について以下の実験③を行うことで各データ拡張手法を品質の観点で評価した。

実験③ 様々なデータ拡張に対して予備実験②の方法で平均識別率と標準偏差を求める

基本的な実験設定は予備実験②と同じである。データ拡張としては変形や回転などの幾何学変換として Skew, Scale Augmentation, Shear, Rotate, Rotate Zoom の 5 種類、ノイズ付加として PCA Color Augmentation, Gaussian Noise, Patch Gaussian, Salt Pepper Noise の 4 種類、色情報の非線形変換として Gamma Transform, Contrast Transform の 2 種類、ぼかし処理として Gaussian Filter と Smoothing Filter の 2 種類、マスク処理として Cutout, Random Erasing, Cut Mix の 3 種類、データ合成として Manifold-Mixup を実装し、各評価を行った。評価の結果を図 3.3 に示す。

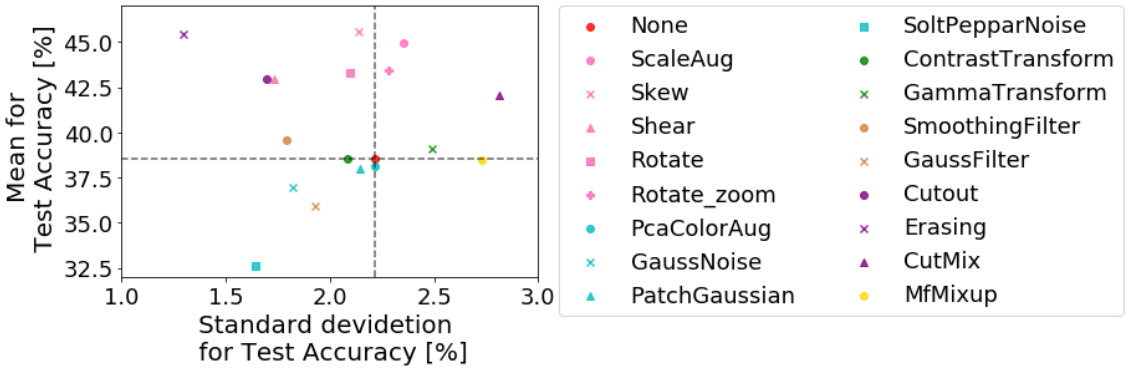


図 3.3 データ拡張による品質評価

それぞれのデータ拡張手法について縦軸に識別精度、横軸に標準偏差をプロットした散布図となっている。データ拡張を何もしていない None（赤点）を基準点として、基準点の左上の第二象限は精度を向上させながら標準偏差が小さくなっているため識別精度を上げながらもばらつきを抑えるデータ拡張であると言うことができ、品質を高める手法であると結論付けることができる。一方で基準点の右上の第一象限は識別精度の向上は認められるものの標準偏差が大きくなっているため、データによっては効果があるが、そのばらつきは大きいと考えられる。基準点の左下の第三象限は認識精度は下がるもののばらつきは小さくなっているため、識別精度は必ずしも上がらないが、品質を保証するという意味では貢献できるデータ拡張手法であるといえる。基準点の右下の第四象限は識別精度を下げつつばらつきを大きくするというデータ拡張に適していない処理がプロットされるエリアであるが、このようなデータ拡張手法は今回の適用例には見られなかった。

従来の state-of-the-art な評価という観点では上に行くほど良いデータ拡張手法であると言えるが、標準偏差を考慮した品質という点で考えると左上に行くほど良いデータ拡張手法であると言える。傾向としては幾何学変換のデータ拡張手法（図中のピンクのプロット）は上へと評価が上がる傾向にあり、ノイズ付加（水色のプロット）は左下へと下がる傾向にある。一方でマスク処理（紫のプロット）は左上へと上がる傾向にあり、データ拡張の中でも特に優れている手法であると評価することができる。

3.3 データ拡張手法の組み合わせの効果

次にデータ拡張としてばらつきを下げる効果のあった幾何学変換とノイズ付加、マスク処理の 3 つについてそれぞれのデータ拡張を全て行う場合とランダムに行う場合に対して一定の確率でデータ拡張する場合と徐々にデータ拡張をする割合を増やした場合を組み合わせて、識別率がどのように変化するかを実験④で検証した。実験④の検証項目は以下の通りである。

実験④ 品質を向上させるデータ拡張手法の組み合わせを明らかにする

幾何学変換、ノイズ付加、マスク処理の 3 つのデータ拡張について、それぞれのデータ拡

張を全て行う場合（All)、ランダムに行う場合（Random)、一定の割合で行う場合（const)、行う割合を線形で増やす場合（Linear）で実験を行い散布図を作成した。図 3.4 にその結果を示す。

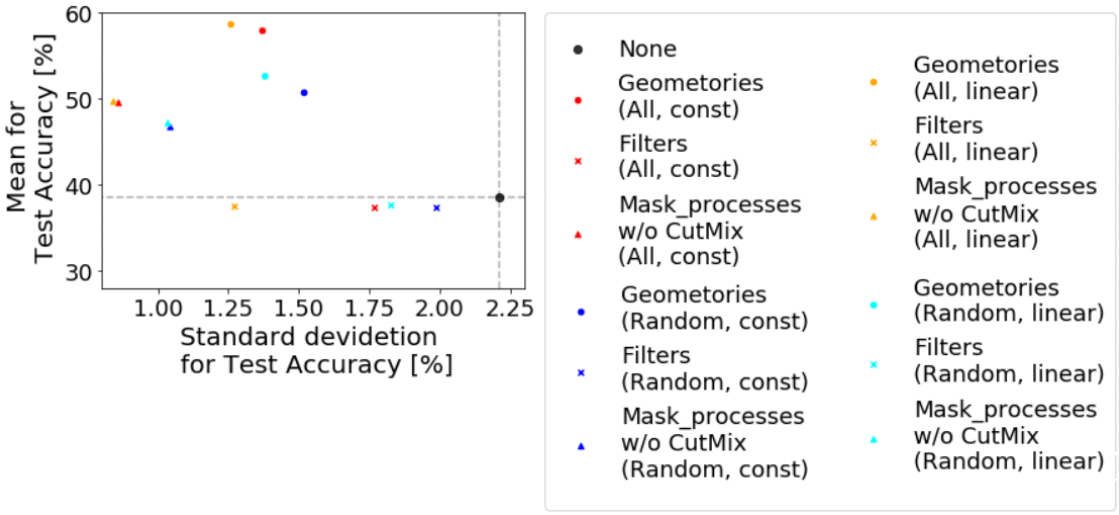


図 3.4　データ拡張手法の組み合わせ実験による品質評価

実験の結果から幾何学変換とマスク処理のデータ拡張においてはランダムに行うよりも全てを行う方が品質がよくなり、一定の確率で行うよりも徐々に行う確率を増やしていった方が識別精度を高く、かつ標準偏差を小さくできることが分かった。一方でノイズ付加に関しては識別精度を上げることはできなかったが、大きく下げることなく標準偏差を小さくする効果があることは分かった。

3.4　まとめ

これまでの多くの論文で行われている識別精度だけで機械学習システムの品質を評価するのではなく、分散や標準偏差も考慮することで品質を評価する必要があることが明らかになり、品質を向上させるためのデータ拡張としてはマスク処理と幾何学変換の組み合わせが有効であることが分かった。

4 データ拡張の適用法の改良による品質改善

本章では、ニューラルネットワーク学習におけるデータ拡張の新しい適用法の開発を行い、実験を通して学習品質への影響を評価した結果について述べる。

4.1 研究目的

データ拡張は、データに変形を加えることでデータ数を増やす技術であり、訓練データ数が少ないときに性能が落ちてしまうという性質をもつ深層学習において、高い効果を発揮する。一方で、データ拡張の有効性は用いるデータに強く依存するため、データ拡張手法の選択や各手法がもつパラメータを適切に設定しなければならない。しかし、データ拡張の理論解析は難しく、汎用的な使用法が確立していないという現状があり、経験的、慣例的な使用がなされるケースが多い。これは意図せず不適切な使用をすることにつながり、学習の品質を損ねてしまう。実際に、各データ拡張手法の変形量、例えばマスクサイズや回転角度などを不適切な値に設定してしまうことにより、学習の性能が落ちてしまうケースや、実際に用いる実データに対して、どのようなデータ拡張手法を選択すればよいのかと頭を悩ませるケースがよく見受けられる。

そこで、データ拡張の経験的な使用からの脱却を目指して、本研究はデータの多様性に焦点を当てた。多様性を高めることはデータ拡張の本質的な目的であり、多様性の増加が汎化性能の向上に大きく影響を及ぼすことは、[20]の研究において実証されている。近年、複数のデータ拡張操作からランダムに選ばれた操作を学習中に動的に適用する RandAugment [21]という手法が注目されているが、これは多様性を大きく向上させる半面、調整が必要なパラメータも多く、効果的に利用するのは容易ではない。本研究では、データの多様性に関連したデータ拡張の適用法として、以下の 2 点を新たに提案し、それぞれのアルゴリズムの改良および性能の評価を行った。

- ニューラルネットワークの中間層を含めた様々な層でデータ拡張を適用し、適用層の自動的な最適化を行う(4.2 節)。
- 有力なデータ拡張手法である Mixup 法[22]を改良し、サンプルの新しい混ぜ合わせ方を提案する(4.3 節)。

4.2 データ拡張の適用層の改良

4.2.1 中間層におけるデータ拡張

一般に、データ拡張は入力データに適用するものであると考えられているが、ニューラルネットワークでは中間層で出力された特徴量を取り出し、データ拡張を適用することが可能である。これに関していくつか先行研究が存在するが、手法を mixup[22]に限定する Manifold mixup[23]や、ほかにも特殊なネットワークやデータを必要とするなど、汎用性の低い手法が多い。本研究では、画像データに対して用いられるアフィン変換やマスク処理といった様々なデータ拡張手法を、中間層で適用することを考えた。CNN では、階層的に特

徴が抽出されるため、ミニバッチごとにランダムに選ばれた様々な層でデータ拡張を行うことで、多様なサンプルが生成される。入力画像への適用と同じように、中間層で得られる特徴マップに対してデータ拡張を適用することができるので、実装も容易である。

実際に入力画像および特徴マップに対して、マスク処理と平行移動を適用した例を図 4.1 に示す。ここでは、学習中のモデルにサンプルを入力し、同じパラメータ（マスク位置、移動量）に設定したデータ拡張を異なる層で適用した直後の画像を、サイズを揃えて上段に表示している。そのサンプルの最終層における特徴マップを下段に示しているが、それらはデータ拡張を適用した層によって異なる画像となっている。この結果から、様々な層でデータ拡張を行うことは、生成データの多様性の向上につながり、入力データのみにデータ拡張を適用する場合とは異なった学習が行われることがわかる。

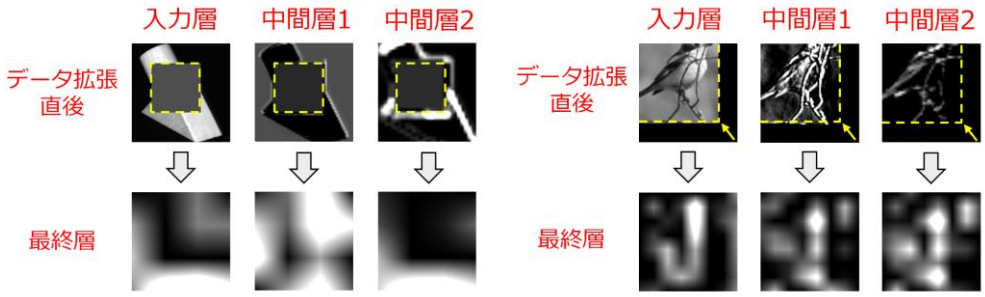


図 4.1　入力画像および中間層で得られる特徴マップにデータ拡張を適用した例

入力層におけるデータ拡張と特徴マップへのデータ拡張の性能を比較するために、様々なデータ拡張を用い、教師ありで学習したモデルのテスト精度を求めた。ここでは、CIFAR-10, Fashion-MNIST, SVHN（補助データを含まない）データセットを用いて、WideResNet28-10 を 200 エポックの間学習した。結果を図 4.2 に示している。各図において、横軸は従来手法（Input DA）、縦軸は提案手法（Latent DA）の精度[%]を表している。これらの結果からわかるように、従来手法よりも提案手法の方が高い精度を示す傾向があり、Crop を用いた結果のように、従来手法では精度が低くなる場合においても、提案手法は高い精度を与えた。この結果から、ランダムな層へのデータ拡張の適用により生成された多様なサンプルは、性能の向上に効果的であることがわかった。

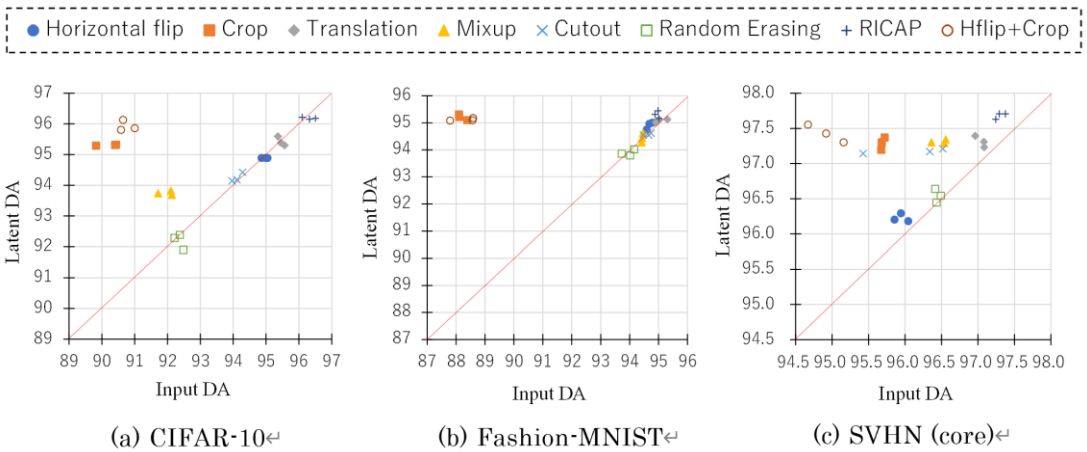


図 4.2　Input DA と Latent DA によるテスト精度の比較

4.2.2 データ拡張に最適な層の選択

中間層でデータ拡張を行うことは効果的であることがこれまでの研究によりわかったが、どの層でデータ拡張を行うのが最適であるのかという問題が出てくる。データ拡張を行う層を変えながら訓練を繰り返し行い、バリデーション精度の値を比較することで最適な層を発見することは可能であるが、全体の訓練時間が増えてしまうため、非効率的な方法であり、実用的ではない。そこで本研究では、1 回の訓練でデータ拡張に最適な層を動的に発見する方法を開発することに取り組んだ。

アプローチとしては、採択率というパラメータを各層で用意し、学習中、採択率の更新を行い、採択率に従って確率的に選ばれた層でデータ拡張を適用する。採択率の更新は、以下に示すような勾配降下法を用いて行う。

$$q_l \leftarrow q_l - \eta \frac{\partial L_{val}}{\partial q_l}$$

ここで、q_lは l 層の採択率、L_{val}はバリデーションデータを入力したときの誤差の値、ηは更新のステップ幅を表す。実際には、バリデーションデータの値をアルゴリズムに含めるべきではないので、データ拡張を加えた訓練データで擬似的にバリデーションデータを作り出し、更新を行っている。学習の初期状態では、すべての採択率を和が 1 になるように均等な値にセットし、ミニバッチごとに採択率の更新を行う。これにより、データ拡張の適した層の採択率は増加し、適さない層の採択率は減少されるといった最適化が行われ、その結果、汎化性能が向上することが期待される。

この手法を Adaptive Layer Selection (AdaLASE)と名付け、従来手法との比較を行った。データは CIFAR-10、モデルは ResNet18 を用い、データ拡張なし、入力にデータ拡張、ランダムな層でデータ拡張、および AdaLASE を用いた場合のテスト精度を比較した。図 4.3 の(a)は Cutout、(b)は Mixup を用いた結果である。初期値を変えた 5 回の精度の平均と標準偏差が、手法ごとに示されている。これらの結果から、AdaLASE は従来手法と同等以上の性能が出せることがわかった。今後の計画として、学習中の採択率の変化を見て、層がどのように選択されていくのか、AdaLASE が正しく機能しているかについて、詳しい解析を行う。

(a) Cutout **(b) Mixup**

図 4.3 AdaLASE と従来法とのテスト精度の比較

4.3 Mixup の改良による新しい混ぜ合わせ方法の提案

実際の訓練において、データ拡張は、例えば切り抜き、回転、および反転のように、複数の手法を同時に用いることが多い。そこで、このように複数の手法を用いるときの、手法間

の相性に着目し、特にデータの多様性という観点から相性を議論することを考える。そのためのアプローチの第一歩として、既存の手法を変形した新しい手法を提案し、元の手法と同時に利用することによって、生成されるデータの多様性を高め、性能の向上を図ることを考えた。多様性を定式化する方法については、[20]の研究において述べられているが、本研究ではまず、精度だけを比較し、提案手法によって性能が向上するかどうかを検証する。

ここでは、データ拡張手法の一つである Mixup[22]の改良を行った。これは、2 つのサンプルの線形補間によって新たなサンプルを生成する手法であり、次式に示されるように、入力値およびラベルのそれぞれについて同じ比率で線形補間をとる。

$$\begin{cases} \tilde{x} = \lambda x_i + (1-\lambda)x_j \\ \tilde{y} = \lambda y_i + (1-\lambda)y_j \end{cases}$$

ここで、(x_i, y_i)および(x_j, y_j)はi番目およびj番目のサンプルの入力値を表し、λはベータ分布からサンプルした混合率を表す。本研究で、Mixup を対象に選んだ理由は、その汎用性の高さにあり、画像だけでなく時系列データなど多くの数値データに利用することができるため、Mixup 法を改良することによる影響が大きいと考えたからである。

Mixup をニューラルネットワークの中間層でも行えるように改良したものは Manifold mixup[23]と呼ばれるが、いずれの Mixup も 2 点間の線分上というデータ分布上の局所的な範囲にしかサンプルが生成されず、またその線分上の点の性質が非線形に変化する分布をもつデータセットに対しては不適切である。

本研究で提案する Feature Combination Mixup（FC-mixup）は、従来の Mixup とは異なる方法でサンプルを混ぜ合わせる手法であり、その概要を図 4.4 に示す。同じミニバッチ内に含まれる 2 つのサンプル A と B が、ランダムに選ばれた層において、Z_AとZ_Bという特徴量を出力するとする。dをその層の特徴量の総数とすると、FC-mixup は、Z_Aから$d\lambda$個、Z_Bから$d(1-\lambda)$個の特徴量をランダムに抽出し組み合わせて新たなサンプルZ_Xを生成する。その組み合わせの数は、一つのλの値について多数考えられるため、乱数に応じて異なったデータが生成され、したがってデータ分布上の広い範囲にサンプルを生成することができる。FC-mixup は次式のように表現されるので、この式が満たされるようにZ_AとZ_Bを混合する。

$$|Z_A \cap Z_X| = d\lambda$$

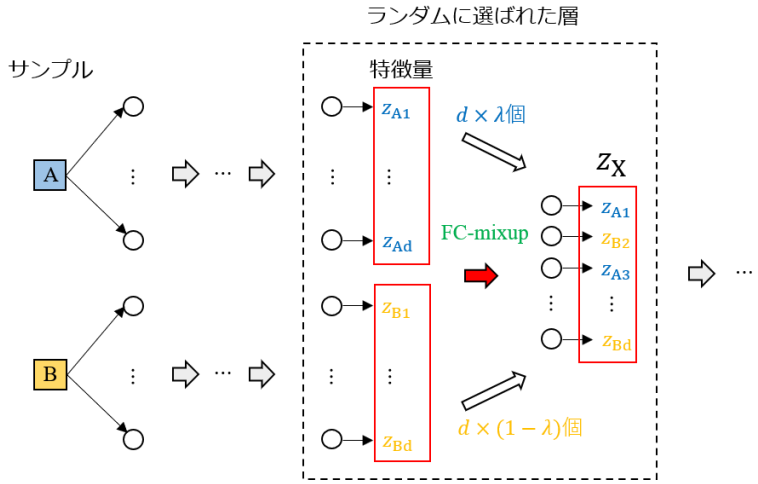


図 4.4 FC-mixup の概要

このように 2 つのデータがもつ各パーツの組み合わせにより新たなデータを生成する技術は、Puzzle Mix[24]においてもみられるが、これは対象が入力画像に限定される。また、Adversarial mixup resysthesis[25]において類似した手法が利用されているが、オートエンコーダでの使用に限られており、FC-mixup はより汎用的な使用を想定して設計されている。生成データの多様性を高めるために、FC-mixup と Manifold mixup[23]を同時に利用する手法を、ここでは Hybrid 法と呼ぶ。

実験では、複数の多クラス分類データセットを用い、従来手法（データ拡張なし、入力層での mixup[22], Manifold Mixup[23]）と提案手法（FC-mixup, Hybrid 法）とで、テストデータの識別精度を比較した。データには、MNIST, CIFAR-10, CIFAR-100, SVHN, および TinyImageNet を用いた。モデルには、中間層を 1 層もつ多層パーセプトロン（MLP)、小さい畳み込みニューラルネットワーク(CNN)、ResNet18、および ResNet50 を用いた。フルサイズのデータに加えて、1,000 サンプルをランダムに抽出した Reduced データでも実験を行った。初期値を変えた 5 回の試行における平均と標準偏差を求め、比較した。

表 4.1 の結果から、多くの場合で提案手法は最も高い精度を示していることがわかる。全体として、Hybrid 法よりも FC-mixup の方が、良い性能を与える傾向がみられた。今回の結果は、FC-mixup や Hybrid 法を試してみることで、品質の改善につながる可能性が高いということが期待される結果であるといえる。多様性に着目したデータ拡張手法間の相性に関する詳しい解析は、今後の課題とする。

表 4.1　多クラス分類データにおけるテスト精度の比較

	MNIST MLP	CIFAR-10 SMALL CNN	CIFAR-10 RESNET18
DEFAULT	98.44 ±0.010	86.76 ±0.34	87.87 ±0.27
INPUT	98.41 ±0.066	87.03 ±0.30	88.47 ±0.35
MANIFOLD	98.55 ±0.044	87.22 ±0.32	88.50 ±0.37
FC	98.70 ±0.050	87.49 ±0.28	88.76 ±0.19
HYBRID	98.62 ±0.020	87.40 ±0.33	88.49 ±0.16
	SVHN RESNET18	CIFAR-100 RESNET50	TINYIMAGENET RESNET50
DEFAULT	93.79 ±1.92	54.97 ±1.58	65.38 ±0.26
INPUT	95.79 ±0.12	60.37 ±1.61	68.27 ±0.60
MANIFOLD	95.73 ±0.08	61.94 ±1.98	68.22 ±0.83
FC	95.71 ±0.12	63.45 ±1.82	68.04 ±0.49
HYBRID	95.68 ±0.16	62.48 ±3.40	69.04 ±0.49
	MNIST (1000) SMALL CNN	CIFAR-10 (1000) SMALL CNN	SVHN (1000) SMALL CNN
DEFAULT	96.24 ±0.12	56.59 ±0.72	68.34 ±1.06
INPUT	96.15 ±0.21	58.44 ±0.86	69.49 ±1.10
MANIFOLD	96.30 ±0.19	56.78 ±0.81	68.51 ±1.36
FC	96.86 ±0.10	59.73 ±0.90	73.82 ±0.59
HYBRID	96.63 ±0.19	58.22 ±0.66	70.47 ±0.64

5 深層 NN ソフトウェアのデバッグ・テスティング

深層 NN ソフトウェア開発の初期段階では 3 つの観点（実現機能の具体化，学習に用いるデータセットの整備，深層 NN 学習モデルの選択）から試行錯誤的な繰り返しを通して，要求機能ならびに予測性能が達成可能かを確認する。この試行錯誤過程は従来のプログラム開発のデバッグ作業に対応するが，深層 NN ソフトウェア（DNN ソフトウェア）の場合，デバッグ・テスティングの入力データセット生成，訓練学習進行状況の監視と評価，要求実現を阻害する原因の特定と除去といった作業になる。以下，令和 2 年度に実施したデバッグ・テスティングの方法を報告し，得られた実験結果の考察と今後の計画を整理する。

5.1 不具合の直接原因

教師あり DNN 学習の標準的な方法では，訓練・学習と予測・推論の 2 種類のプログラムが関わる。学習データを与えられて学習タスクが決まった時，目的とする DNN ソフトウェアの実現に必要な学習モデルを選び，また，訓練学習過程で用いる方式を決める。利用可能な OSS の学習フレームワーク提供機能を用いる場合，フレームワークのパラメータを決めれば良い。次に，学習データから訓練データセットを構築する。そして，学習モデル・訓練データセットを入力として訓練・学習プログラム（学習フレームワーク提供）を作動させ，その結果，訓練済み学習モデルを導出する。より詳細には，訓練・学習プログラムが求めるのは，訓練済み学習モデルを定義する重みパラメータ値の集まりである。この訓練済み学習モデルが予測・推論プログラムの振舞いを規定する。

利用者からみると，DNN ソフトウェアの実体は予測・推論プログラムである。たとえば分類学習タスクの場合，入力データに対する分類の確からしさを求めるプログラムである。そして，この出力結果を調べることで，構築した DNN ソフトウェアが意図通りに作動しているかを判断する。期待する結果が得られず不具合があるとみなす時，訓練・学習プログラムの実行以前にもどって，欠陥の在り処を調べて除去する。つまり，デバッグ作業を行う。

不具合が生じる時，訓練・学習プログラム実行過程で用いる情報の何処かに欠陥があり，訓練データセット・学習モデル・学習機構のいずれか，あるいは，これらの複数が原因となる。一方，予測・推論結果に不具合をもたらす直接原因は，訓練済み学習モデルあるいは重みパラメータ値の集まりである。訓練データセット・学習モデル・学習機構の欠陥が不具合の根本原因である一方で，直接原因は重みパラメータ値にある。つまり，根本原因となった欠陥は重みパラメータ値の不具合として顕在化し，その不具合が示す訓練済み学習モデルの歪みが利用者からみた不具合の直接原因となる欠陥である[26]。この歪みを計測する方法が必要である。

本章では，重みパラメータ値を測定する内部指標を導入することで，DNN ソフトウェアの不具合を検知できるか否かを調べる。重みパラメータ値は訓練・学習プログラムの出力であるが，その出力値が妥当かを調べる直接の方法はない。その理由は，出力として期待する重みパラメータ値を予め知ることができないことによる。このような期待する重みパラメータ値が既知であれば，訓練・学習は不要になる。その既知の値を使えばよい。

5.2 内部指標

ニューロン・カバレッジ（NC）の考え方を紹介する。学習モデルをニューロンのネットワークとみなす。閾値を決めた時，出力値が閾値を超えるニューロンは活性状態にあるという。学習モデルを構成するニューロン数を N とし，活性状態のニューロン数を A とする時，活性ニューロンの比（NC = A/N）をニューロン・カバレッジと定義する。文献[27]は，NC を検査網羅性の基準と仮定し，評価用入力データの選び方が NC 値，すなわち，訓練済み学習モデルの検査網羅性に影響することを調べた。

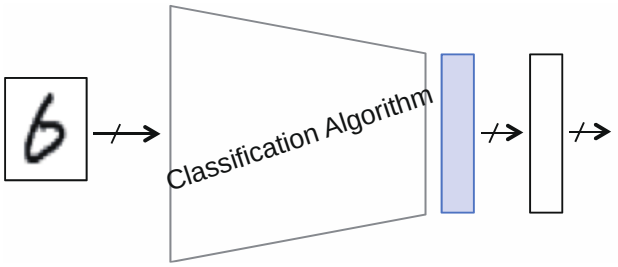


図 5.1 訓練済み学習モデル

本章では，NC を計算する対象ニューロンの選び方を工夫することで，不具合の有無を調べる内部指標[28]として用いる。図 5.1 に訓練済み学習モデルの模式的な図を示した。中間層の最終段階（網掛けした層）を対象とするニューロンに対して NC を定義し内部指標とする。

一般に，機械学習の技術では，此の Penultimate Layer を特別の観測対象とすることが多い。たとえば，画像分類タスクの場合，その前段までが画像認識などの具体的なアルゴリズムの役割を果たす相関分析（ピクセル値のパターンの分析）の処理であり，その計算結果が，此の層に集約されることが理由である。そして，本章では，想定される欠陥原因が，この内部状態として顕在化すると仮定する。さらに，この内部状態をもとに，さまざまな統計指標を導出することができる。調べたい不具合によって，何が適切な導出指標かを実験によって調べる。

5.3 実験の方法と結果

いくつかの実験結果を示し，先に定義した内部指標あるいは導出指標の有用性を考察する。最初に，訓練・学習プログラム（学習フレームワーク）に欠陥がある時の比較実験結果を示す。以下，BI は PC に欠陥挿入した訓練・学習プログラムである。

図 5.2 は学習モデルとして古典的な全結合ネットワークを用い，中間層のニューロン数を変化させて，試験データセットの正解率をプロットした。十分な数のニューロンを持つ時（横軸で 50），PC と BI で正解率に大きな差がないことがわかる。つまり，正解率を調べても，PC と BI を区別することが困難であり，その結果，欠陥の有無がわからない。これは，これまでに得られた知見を再確認するものである。以下，この知見（図 5.2）に加えて，さらに状況を系統的に調べる実験の結果（図 5.3）を示す。

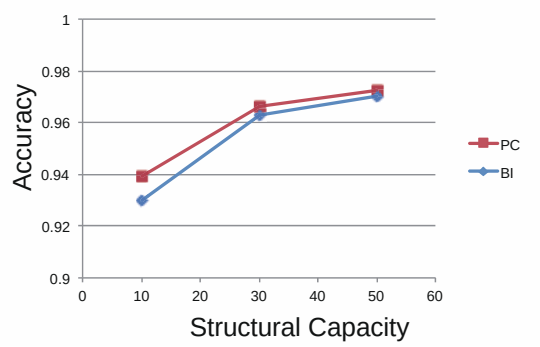


図 5.2 中間層の異なる学習モデル

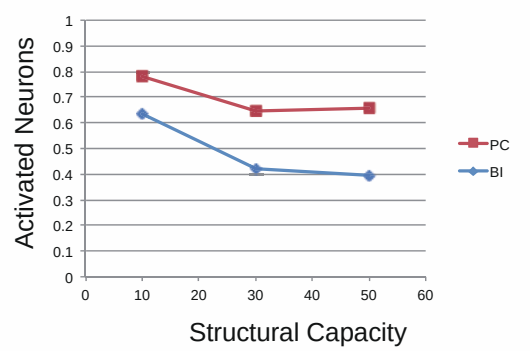


図 5.3 内部指標との関係

図 5.3 は，縦軸に内部指標（本章でのニューロン・カバレッジ）をプロットした。この指標の絶対値を参照すると，たとえば、BI の 10 と PC の 30 とは，共に 0.7 程度であって，BI と PC の区別がつかない。そこで, 内部指標からの導出指標に適切なものがあるかを調べる。今，試験データセットに対するニューロン・カバレッジの集まりを得て，この平均値 μ と分散 σ^2 を求めて，さらに，σ / μ を計算する。横軸にこの導出指標 σ / μ を用いる場合を図 5.4 に示した。縦軸の値から図 5.3 を参照することで，どの学習モデルがどの値かを知ることができる。

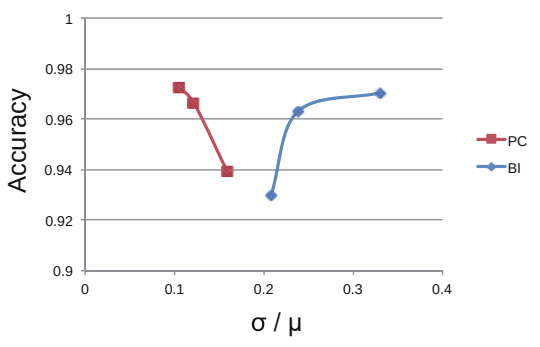


図 5.4 導出指標

図 5.4 によると，PC と BI を区別できることがわかる。つまり，内部指標では，キャパシ

ティ（中間層のニューロン数）が異なる PC と BI を区別できない（図 5.3）が，導出指標を工夫して σ/μ によって比較すると，ニューロン・カバレッジが有用な情報を与えることがわかる。

次に，欠損データを評価用に用いて，PC と BI が個々のデータに対して出力する分類の確からしさを散布図（図 5.5）で表す。

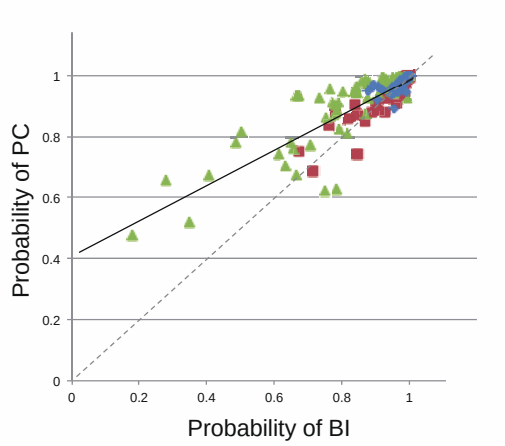


図 5.5　分類の確からしさ

図 5.5 で△が欠損データに対する出力値を表す。PC と BI で同等の値を出力すると仮定すると，原点を通る点線上に分布する筈である。実際，試験データセットから選んだ□は概ね此の線上にのることがわかる。一方，欠損データ（△）は実線上に分布し，PC がより良い分類の確からしさであること示す。つまり，欠陥混入した BI は、正解率は変わらない（図 5.2 を参照）が，頑健性に劣るといえる。

次の実験は，内部指標を用いることで頑健性の違いを検出できることを確認するものである。

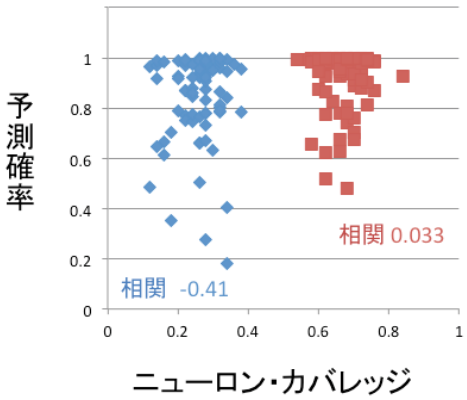


図 5.6　内部指標の違い

図 5.6 は，前記の欠損データを入力し，先に定義した内部指標を横軸にプロットした。右側に分布する□は PC，左側に分布する◇は BI による結果を示す。これによると，(1) PC の内部指標の値が大きいこと，(2) 内部指標と予測確率（分類の確からしさ）の相関が弱いこと，がわかる。次に，σ/μ を計算すると，PC は 0.0876 であり，BI は 0.2183 となった。図

5.6 は頑健性に影響する欠損データを用いる実験であり，σ/μ の値が頑健性と強く関係すると考えられる。

次に，訓練データセットに系統的な歪みを与えて，訓練済み学習モデルを導出する実験を行った。系統的な歪みが生成できることが，これまでの実験からわかっている。この実験は，訓練データセットの違いが，内部指標に影響を与えるかを調べることに相当する。同一の試験データセットに対する正解率をプロットした。

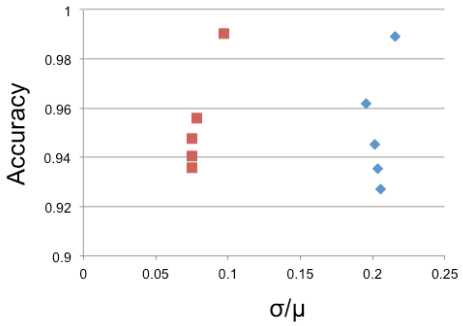


図 5.7　訓練データセットの違い

図 5.7 の上から下へ（正解率の良い方から悪い方へ），大きい歪みの訓練データセットを用いた場合に対応する。つまり，試験データセットのデータ・シフトが訓練時に用いたデータセットから相対的に大きくなるので正解率が低下することを確認できる。一方で，横軸の値（σ/μ）は，PC（□）と BI（◇）で明らかに異なる。この実験の状況は，訓練データセット・シフトの一例と考えて良い。図 5.7 は PC と BI で独立した 2 系列を示す。正解率が示す正確性と σ/μ 値が示唆する頑健性とが独立な観点であることを確認できる。

以上から，訓練データセットの歪みは，内部指標では区別がつきにくいが，正解率に基づく方法で調べることができる。実務で行われているように，訓練データセットの良し悪しの検討に際しては，正解率を調べる方法が有用であると云えよう。一方，訓練・学習プログラムの欠陥など他の要因が関わる可能性がある（多重欠陥が想定される）場合，内部指標や導出指標（σ/μ）の値を同時に調べることが望ましい。

5.4　関連研究

ニューロン・カバレッジは DeepXplore [27]で導入されたメトリックスである。従来のソフトウェア・テスティングで用いられてきたテスト網羅性基準を参考に提案された。従来法は，実行文をノードとする制御フロー・グラフ（Control Flow Graph, CFG）でプログラムを表現する時，あるテスト入力データによって実行される文を基本単位として，テスト網羅性を定義する。最も簡単には，CFG のノードが入力テストによって実行される経路に含まれるか否か，つまり文が実行されるか否かを基準とする。これは，文網羅基準あるいは C0 基準と呼ばれる。DNN 学習モデルはネットワークとして表現されることから，CFG 上で定義された C0 基準との対比により，ノードに位置するニューロンが活性化（出力値が閾値を超えること）されるか否かによって，ニューロン・カバレッジ（NC）を定義した。そして，NC 値を増加させるように新しいテスト入力データを生成する方法を論じた。

ニューロン・カバレッジは従来のテスト網羅性基準から類推できる素直な考え方だろう。その後，NC を向上させる入力データの作成が難しくないという経験から，複数ニューロンの相関や異なる層の間での相関を考慮したメトリックスが提案された[29]。また，NC が増加するように，NC 値をガイドとして，データ補完の方法で有用なテスト入力データを系統的に生成する方法[30]がある。さらに，NC をガイドに用いる方法と GAN をベースとするテスト入力自動生成[31]とを組み合わせる方法が論じられている[32]。データ生成をガイドする指標として，NC が有用なことが確認されたと言ってよい。

テスティングの例として，文献[30]および[31]は回帰問題 DNN モデルの予測結果をもとに計算したステアリング角度というアプリケーションの機能を検査の観点とした。文献[33]は NC 値を大きくするテスト入力が欠陥発見に役立つかを調べた。何を欠陥とするかで NC が有用か否かの判断が異なることを論じている。逆に，この研究[33]は，正確性を中心とする外部指標と NC の間の相関が弱いことを述べている。本章では，両者に相関が弱いという観察から，NC に基づく内部指標を検査に用いた。文献[33]と矛盾しないどころか，同じ方向の議論を展開しているといえる。なお，網羅性は検査終了の判断基準であり，一方，欠陥発見はテスト入力データがコーナーケースを実行するかに依存する。両者は異なる側面を論じているともいえる。実際、従来のソフトウェア・テスティングにおいても，網羅性の向上が必ずしも欠陥発見の効率向上に結びつかないことが報告されている。

本章の方法は，文献[26][28]で論じられているように，NC 値を簡易的な検査指標に用いるというものである。従来の研究が NC を検査の網羅性基準に用いるのに対して，DNN モデルの欠陥が NC 値として現れる，という見方を採用した。実験では，この考え方に基づく具体的な検査として，訓練・学習プログラムの信頼性および訓練済み学習モデルの頑健性を調べることができた。

5.5 おわりに

本章では，Penultimate Layer でのニューロン・カバレッジに基づく内部指標を用いた。これはスカラーであることから，計測ならびに導出指標の定義が容易であり，検査指標として利用しやすい。一方，NC はニューロン個々の値に関する情報を捨象しており，有用な情報が欠落する。実際，文献[34]では，ニューロン値の分布を推定し，これをもとにテスト入力データが妥当かを論じる方法を提案している。これを応用すると，ニューロン値の分布は訓練データセットの歪みを，より詳細に表すと考えられるだろう。今後，この分布を利用する考え方を応用することで，訓練データセットをデバッグする方法を検討する。

6 訓練データのデバッグ・テスティング

6.1 ３つの問題設定

深層ニューラル・ネットワーク（Deep Neural Networks、DNN）の技術[35]を応用した DNN ソフトウェア開発初期には、中核となる DNN コンポーネントの機能振舞いが期待通りであるかの確認を目的としたデバッグ・テスティングを行う。これは DNN コンポーネントに適切なデータを入力し、予測出力が意図通りかを調べる作業である。出力に何らかの不具合がある時、検査対象の DNN コンポーネントが欠陥を含む。デバッグの目的は、このような未知の欠陥を特定し除去することである。

DNN コンポーネントの欠陥は不具合の直接原因であるが、根本原因ではない。DNN コンポーネント構築の標準的な方法[36]では、(a) 学習基盤、(b) 学習モデル（DNN モデルの雛形)、(c) 訓練データ、といった 3 つの要素が複合的に関わる。その何処かに DNN コンポーネントが示す不具合を導いた根本原因がある。何処に根本原因を想定するかで検査の問題設定が異なる[37]。

DNN コンポーネント構築の基本は、膨大な数の学習データからなる訓練データセットを対象とし、訓練データに内在する情報を統計的な方法を用いて帰納的に導出して DNN モデル（非線形関数）を求めること。素朴には、DNN モデルを調べて根本原因を特定すれば良い。しかし、DNN モデルは非線形関数で表現した処理手順なので、DNN モデルに評価データを入力し機能振舞いあるいは出力結果が妥当かを調べる間接的なテストオラクルによるソフトウェア・テスティング法を用いる[38]。

先の(a)の場合、学習基盤の実体は最適化問題を解く数値計算プログラムで、メタモルフィック・テスティング検査法が有用なことが知られている[39]。(b)の場合、学習モデルに明らかな欠陥があるわけではない。対象学習タスクに対する最良の学習モデルを見つけることであり、DNN 技術を応用する研究の主要な課題である[35]。本章では、(c)の場合について、つまり、訓練データのデバッグ・テスティングの方法を検討する。

6.2 訓練データのデバッグ問題

訓練データのデバッグ・テスティングは、学習データの偏りが DNN モデルに影響するという観察に基づき、意図通りの機能振舞いを示す DNN モデルが得られるように訓練データを改訂（追加・削除）することである。以下、教師あり分類学習タスクを対象として、訓練データのデバッグ問題を具体的に考える。

6.2.1 モデル正確性とモデル・ロバスト性

入力データをC種類に分類する教師あり学習タスクでは、多次元ベクトルxと正解タグyからなるデータ点z（$z = \langle x, y \rangle$）を考える。そして、与えられた訓練データセットS（$S = \{z^{(k)} \mid k = 1, \ldots, N\}$）から導出された DNN モデルを検査対象とする。DNN モデルは入力の評価データxに対応するC次元の分類確率ベクトルP_xを返す。P_xのj成分を$P_x[j]$として、値が最大の成分jがyと一致（$y = \text{argmax}_{(j \in [1,C])} P_x[j]$）すれば正解とする。この時、多次元

ベクトル P_x、特に、正解の成分 $P_x[j]$ が示す確率は、データ x に対するモデル正確性を表す 1 つの指標となる。また、評価データの集まり E（$E = \{\langle x^{(\ell)}, y^{(\ell)}\rangle \mid \ell = 1, \ldots, M\}$）に対する正解の度数（正解率）を正確さ（Accuracy）とし、正確さもモデル正確性の指標のひとつである。さらに、C種類の分類カテゴリの正確さが不均等か否かもモデル正確性を表す指標となる。

DNN モデル構築過程では、訓練データセット S から求めた正確性と S とは異なるデータセット E に対するモデル正確性を求める。S に対しては良い正解率を示す一方で、E に対しては正解率が悪化することがある。これは訓練データセットに過適合しやすいという現象として知られている。通常、S と E はひとつの大きなデータ・プール D から選んで構成することから、同じデータ分布に従う異なる標本と考える。S と E に対する正解率を調べることで S への過学習の程度を調べる。過学習が見られない時、DNN モデルは汎化性能に優れているとする。

訓練データのデバッグ問題では、評価データ E を D 以外から選ぶこともある. たとえば、期待する振舞いを示すことの確認が目的の正常系テスティングでは、汎化性能の評価と同じように、D から S と異なる E を選べば良い。しかし、例外的な状況での振舞いを調べるには、D に含まれないデータを評価用データセット F としたい。F に対しては、正解率に代表されるモデル正確性は良い指標にならない。F の特定データに対する予測確率（モデル正確性の指標）からの判断ではなく、D のデータからの外れ具合に応じた予測確率の低下が許容範囲であるかを表すモデル・ロバスト性が評価の基準となる。

なお、実際の開発時においては、所与の D で期待する予測性能が得られない場合、新たなデータを収集し、訓練データそのものを改訂する。そして、新たな訓練データセットを用いて、DNN コンポーネントを導出する。検査時には、正常系ならびに例外系テスティングの双方から、モデル正確性とモデル・ロバスト性を評価する。学習モデルを当初の D に限定することは実務に合わないことがわかる。

6.2.2 訓練データ記憶

過適合や過学習は DNN モデルの予測性能（モデル正確性とモデル・ロバスト性）に大きく影響する。そこで、過学習の問題を緩和する方法を組み込んだ学習基盤の方式が従来から研究されてきた。正則化やドロップアウトといった手法である[40]。このような方法を学習基盤が持つ場合であっても、訓練データセットに偏りがあると、期待する予測性能を得ることができない。訓練データのデバッグ問題は、訓練データセットを改訂することで、DNN モデルの予測性能を改善することである。簡単には、不適切な偏りをなくすことと言える。しかし、偏りの度合い、ならびに、偏りの適切さ（不適切さ）を評価することが難しい。

訓練データ（標本）の偏りを評価する方法として、標本の統計的な特徴を調べるアプローチがある。たとえば、$S = \{\langle x^{(k)}, y^{(k)}\rangle \mid k = 1, \ldots, N\}$ とする時、正解タグによってC個の集まり $S^C = \{\langle x, c\rangle \mid \langle x, c\rangle \in S \text{ かつ } c = 1, \ldots, C\}$ に分けたとしよう。S^C の大きさが均等であれば、正解タグからみた偏りはないとしてよい。しかし、各々の S^C は何らかのデータ分布 ρ^C に従うし、この ρ^C から見て偏りがあるかは何もわからない。これを調べるには、ρ^C を推定する必要がある。しかし、データ x は多次元ベクトルであり、このような多次元データ分布 ρ^C を推定することは容易でない。

DNN モデルの予測性能は、評価データを入力してのテスティング結果によって調べる。学習基盤の方式によって、同じ訓練データから導出した DNN モデルが異なる予測性能を示すこともある。つまり、訓練データのデバッグを目的とする検査では、訓練データの統計的な特徴を調べるだけでは不十分であり、訓練データの偏りが、どのように DNN モデルに反映されているか、どのような方式で訓練学習されたか、を考慮する必要がある。

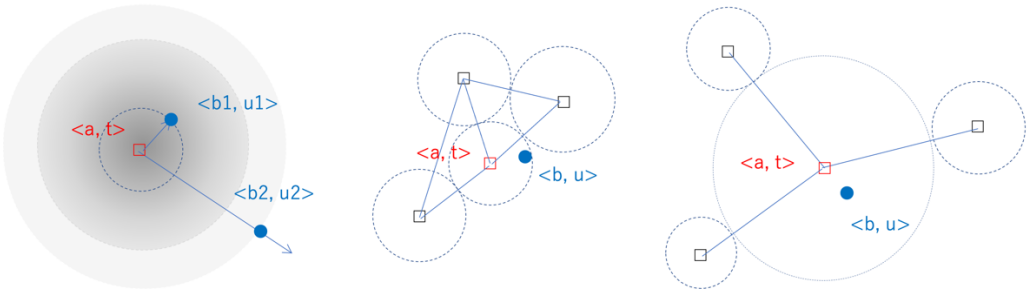
訓練データの偏りと DNN モデルの関係は、DNN モデルが訓練データの教師ラベルを記憶するという観点から論じることができる。今、訓練データ S がデータ点 $\langle a,t\rangle$ を含む時（$\langle a,t\rangle \in S$）、S から $\langle a,t\rangle$ を除去した訓練データを S' とする（$S' = S \setminus \{\langle a,t\rangle\}$）。$S$ あるいは S' を用いて訓練することで得られる DNN モデルを各々 M, M' とする。そして、M による入力 a に対する予測確率ベクトル P_a と M' による P'_a を求める。分類結果 t に対する $P_a[t]$ の確からしさが大きい一方、$P'_a[t]$ に対する確からしさが小さい時、M は訓練に用いたデータ $\langle a,t\rangle$ を記憶する（Memorize）という。この定義から、過適合の状況では、DNN モデルが訓練データを記憶することがわかる。

DNN モデルでは、帰属関係推定（Membership Inference）が可能なことが知られている。データ点 $\langle x,y\rangle$（$\langle x,y\rangle \in D$）が訓練データセット S に含まれていたか（$\langle x,y\rangle \in S$）を訓練済み学習モデルM^Sに入力して得られる情報から調べる問題である。$M^S(x)$ の実行結果の分類確率ベクトル P_x から調べるブラックボックス法[41]や $M^S(x)$ の実行過程で計算する損失関数 $\ell(Y(W;x),y)$ の情報を利用するホワイトボックス法[42]がある。ここで、W を学習パラメータあるいは重みとし、$Y(W;x)$ を入力 x に対する予測の内部表現値とした。なお、これらを用いると、M^S は訓練データセット S の分布 ρ の下で、損失関数の平均値を最小化する問題の解 W^*（$W^* = \mathrm{argmin}_W\, \mathbb{E}_{\langle x,y\rangle\sim\rho}[\ell(Y(W;x),y)]$）から定義する$Y(W^*;x)$ から得られる。

直感的には、帰属関係推定の方法は、データ点 $\langle x,y\rangle$ が訓練データセット S に含まれているか否かによって、P_x あるいは $\ell(Y(W;x),y)$ の分布が異なるという観察に基づく。このような指標の分布に違いが生じる原因は過適合を含む訓練データ記憶である[42]。そして、帰属関係推定の脅威を緩和するアプローチは、過適合が生じないような学習基盤の方式を利用することに加えて、記憶されやすいデータを訓練データセットから除去することである[43]。

ここで、訓練データ記憶と関わる状況を調べる。分類問題を考えて図 6.1 で、$a \neq b$ であるが $t = u$ とする。図 6.1 (a) は訓練データ $\langle a,t\rangle$ から離れると共にデータ $\langle b,u\rangle$ の予測確率が低下することを模式的に示す。図 6.1 (b)は S の中で訓練データ $\langle a,t\rangle$ が密集している時、そのデータを除去してもデータ $\langle b,u\rangle$ の予測確率が大きな影響を受けないことを表す。つまり、除去した訓練データは予測結果に大きな影響を与えないという点で記憶されない。図 6.1 (c)は訓練データが疎な時、図 6.1 (b)とは逆に、影響が大きくなり、強く記憶されていることを表す。このような外れ値データは DNN モデルの予測分類性能に大きく影響する。

最後に、帰属関係推定を訓練データ・デバッグの問題から考える。訓練データが記憶される状況では、データ点が訓練データセット S に含まれるか否かで、P_x や $\ell(Y(W;x),y)$ の分布が大きく異なる。帰属関係推定の方法は、訓練に用いた z から遠いデータ点 z' に対する予測性能が悪いことを利用している。つまり、帰属関係推定をモデル・ロバスト性の検査と考えることもできる。訓練データ記憶の現象はモデル・ロバスト性と関わる。



(a) 近傍での予測確率　　(b) 密集する領域　　(c) 疎な領域

図 6.1　訓練データの配置と予測の確からしさ

ここで、図 6.1 の模式的な状況を参照する。図 6.1 (b)に示す密集データを除去してもモデル正確性への影響は少ないだろう。一方、図 6.1 (c)のようなデータを除去するとモデル・ロバスト性を改善するが、予測分類をサポートするデータがなくなることから、その近傍のモデル正確性が低下する。あるいは、このデータを除去しないで、その近傍に新たなデータを追加すれば密になり、局所的なモデル正確性が向上する。したがって、訓練データ・デバッグでは、訓練データセット S の外れ値を検知することが大切である。

図 6.1 の模式的な説明は、入力データの予測分類性能が訓練データとの位置関係に影響されることを示す。ところが、どのようにして位置関係を定義するのか、つまりデータのどのような特徴から位置関係を定義するのか、を述べていない。逆に、予測分類性能の違いを議論する際に、どのように位置関係を定義すべきなのかという問題ともいえる。位置関係の基準を決めることで、外れ値検知の問題があきらかになる。

6.3　外れ値とニューロン・カバレッジ

訓練データ・デバッグを目的とする外れ値検知の方法を考察する。

6.3.1　訓練データの外れ値

訓練データのデバッグ問題は、訓練データセットから外れ値（Outliers）を見つけ出し、開発対象 DNN モデルの目的に応じて、外れ値の取扱いを決めることである。外れ値を除去する、外れ値周辺に新たなデータを追加する、などの方策が考えられるが、その取り扱いは、DNN モデルに対する要求仕様と関わる。訓練データ・デバッグの一般的な議論としては、外れ値検知の技術を確立することである。

一般に、外れ値は多数を占めるデータと異なる特徴を持つデータであり、外れ値であるか否かは対象データの集まりが示すデータ分布（統計的なデータ・モデル）を前提として定義する[44]。たとえば、データ分布の確率密度関数が既知であれば、データ $\boldsymbol{x}$ の尤度をもとに外れ値であるかを調べればよい。

素朴には、訓練データの経験分布をもとに、外れ値であるかを考える。ところが、訓練データは多次元ベクトルであり、経験分布を簡便な形式で知ることが困難である。たとえば、カーネル密度推定（Kernel Density Estimation）などの手法の適用が難しく、その結果、尤度

に基づく外れ値の検出は実用的ではない。あるいは、ソフトウェア・テスティングの分野で知られている組み合わせテスト法（Combination Testing）に準じる分析方法を応用することがある。経験分布に大きな影響を与えると思われる成分（因子）を選び、このような代表的な次元に着目して調べることで経験分布に対する分析を代用する。実務的に有用な一方、外れ値は元来の定義から稀なデータであり、この近似的な方法の有効性に疑問が残る。

少し視点を変えて、モデル・ロバスト性を分析する標準的な方法のロバスト半径[45]を考える。2 つのデータ点 $\langle x, y\rangle$ と $\langle x', y'\rangle$ を選び、各々の出力の予測分類結果を $P_x[y]$ と $P_{x'}[y']$ とする。ロバスト半径 δ は、出力の違いの許容水準 ε が与えられた時、ε を満たす入力データの違いの最大値 δ（$|P_x[y] - P_{x'}[y']| \leq \varepsilon$ の時、$|x - x'|_p \leq \delta$）である。ここで、入力データの違いをL_pノルムで定義する。素朴には、与えられた ε に対して、ロバスト半径 δ が大きいとモデル・ロバスト性が良いと考える。しかし、ノルムL_pの選び方によってロバスト半径 δ_p が異なる。ロバスト半径によるモデル・ロバスト性の定義は厳密である一方、入力データの空間での分析は、用いたノルムが妥当かといった議論あるいは解釈を必要とし、問題の状況が複雑化する。

ここで、異なる分類カテゴリーのデータ $\langle b2, u2\rangle$ への訓練データ $\langle a, t\rangle$（$t \neq u2$）の影響について考える（図 6.1 (a)）。2 つは分類カテゴリーが異なるので、入力データ空間では離れているとして良い。訓練データ記憶の議論と同様に、訓練データセット S が $\langle a, t\rangle$ を含み、S から $\langle a, t\rangle$ を除去した訓練データ S' とする。各々から得られた DNN モデルを M, M' とし、データ $\langle b2, u2\rangle$ に対する予測分類結果をP_{b2}, P'_{b2} とする。S と S' の誤差関数への影響を分析する影響関数（Influence Functions）の方法によって、$P_{b2}[u2]$ と $P'_{b2}[u2]$ の確からしさが異なる $\langle b2, u2\rangle$ が存在することがわかる[46]。つまり、訓練データ $\langle a, t\rangle$ の有無が $\langle b2, u2\rangle$ 予測分類確率に影響することを表す。したがって、入力データ空間での違いによる分析（$a \neq b2$）だけでは目的とする情報を得ることが難しい。

以上から、訓練データの集まりを対象として、つまり、入力のデータ空間での分析によって、目的とする外れ値を系統的に検知することが難しいとわかる。その理由は、モデル正確性やモデル・ロバスト性が訓練データだけではなく、学習方式など訓練学習に関わる多様な要素に影響されるからである。ただし、入力データ空間での分析が全く効果ないことを主張するものではない。このような分析によって、訓練データの経験分布を大まかに把握することができるだろう。

本章では、入力データ空間の特徴がモデル正確性やモデル・ロバスト性と関わるとしても、系統的な訓練データ・デバッグの方法としては不十分と考える。訓練データの外れ値を検知する系統的な方法を検討する。

6.3.2 活性ニューロン

ニューロン・カバレッジは、対象ニューロン数に対する活性ニューロンの割合で定義される[47]。ここで、$M^S(x)$ とした入力データ x の信号が DNN モデル中を伝播し、各々のニューロンを活性化すると考える。与えられた閾値を超える出力がある時、活性ニューロンと呼ぶ。

ニューロン・カバレッジは、当初、カバレッジ駆動テストデータ生成（Coverage-driven Test Data Generation）の網羅性基準として提案された[47]。入力データ x による活性ニューロン

は出力結果に影響を与えるという点で有用な働きを示す。一方で、入力データ x に対しては予測推論に関わらない時、不活性ニューロンが生じると解釈する。そして、不活性ニューロンを生じる入力データでは、全てのニューロンを調べていないと考え、十分なテストができていないとし、不活性ニューロンを活性化させるような新しい入力テストデータを合成する。

文献[13]による提案の後、ニューロン・カバレッジのテスト網羅性基準としての有用性などに関する研究が進められた[48][49][50]。特に、ソフトウェア・テスティングの C0 基準がそうであるように、ニューロン・カバレッジを達成することは難しくなくテスト網羅性基準として弱いということがわかっている。

一方、そもそも訓練データに不備があり、予測推論に関わらない不活性ニューロンが生じたと解釈することもできる。この場合、新たな入力として得られたデータを訓練データに追加して再学習する[47]。つまり、ニューロン・カバレッジを M^S のモデル品質を評価する基準とする考え方を示唆する。以下、モデル品質評価基準としてのニューロン・カバレッジという観点[51]から考察する。

分類学習の DNN モデル M^S は入力に近い上流層が符号化 E'（Encoding）を行い、その後に分類 C'（Classifying）を担うという流れを示す（$M^S = C' \circ E'$）。ここで、_∘_は関数結合を表し、$M^S(x) = (C' \circ E')(x) = C'\big(E'(x)\big)$ である。出力を分類確率ベクトルとする場合、C' の logits と呼ぶ出力の最終層に softmax 関数を配置する。$M^S = \mathrm{SOFTMAX} \circ C \circ E'$ のように表現できる。次に、E' と C をつなぐ層に全結合ネットワーク（Fully Connected Network）FCN を配置し、$M^S = \mathrm{SOFTMAX} \circ C \circ \mathrm{FCN} \circ E$ に変形する。FCN 層に符合化処理 E の結果が移され、さらに C に伝播されると考える。

ニューロン・カバレッジを計算する場合、閾値の選び方ならびに対象ニューロンの決め方が問題となる。FCN は全結合なので同一層内でのニューロンの入れ替えに対して出力が保存されるスワップ不変性（Swap Invariant）が成り立つ。そこで、 FCN 層のニューロン活性状況を要約する情報としてニューロン・カバレッジが有用だろう。一方、SOFTMAX や CNN のように構成ニューロンが特定の機能的な役割を担う場合、すべてのニューロンを同等に考えるニューロン・カバレッジが有用な情報を提供するかは疑問がある。実際、CNN に対しては、2 つの異なる定義が知られており、どちらを採用するかで、ニューロン・カバレッジの値が異なる[52]。本章では FCN 層に対してニューロン・カバレッジを考える。

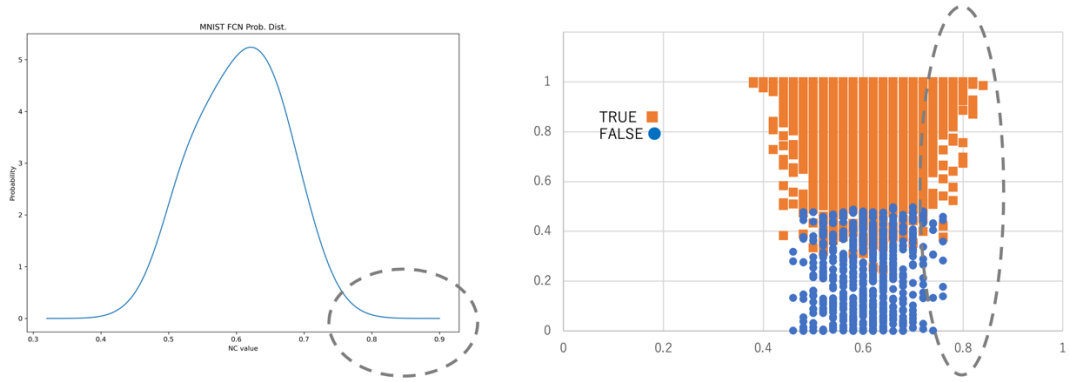
文献[52]は、データ補完の方法で学習データを系統的に変化させて、E' や C でのニューロン・カバレッジへの影響を調べた。その結果、訓練データの違いが E'層のニューロン・カバレッジに大きく影響する一方で C層への影響は小さいこと、評価用の試験データの違いは C の最終層（Penultimate Layer）にほとんど影響しないことなどがわかった。先に導入したように $E' = \mathrm{FCN} \circ E$ とする時、この FCN 層に訓練データの違いが反映されると考えてよいだろう。

また、これまでの実験[39][51][53]において、C の最終層に設けた FCN のニューロン・カバレッジを測定したところ、分類予測確率とニューロン・カバレッジの相関がほとんどないことがわかった。したがって、ニューロン・カバレッジはモデル正確性に寄与する情報と独立した側面を表すと考えられる。仮にモデル・ロバスト性との間に何らかの相関関係があるとしたら、ニューロン・カバレッジが外れ値を検知する方法として有用と期待できる。

6.4 実験と考察

実験では、$M^S = \mathrm{SOFTMAX} \circ C \circ \mathrm{FCN} \circ E$ の形を想定し、MNIST データセットを用いた。検査対象の訓練データセット S を入力し FCN 層でのニューロン・カバレッジを測定した。図 6.2 は活性ニューロンによって訓練データを評価した時の分析結果である。

図 6.2 (a)は横軸に入力データに対するニューロン・カバレッジ、縦軸に対応するニューロン・カバレッジ値を得た入力データの度数分布（KDE の結果）を表す。図 6.2 (b)は横軸に入力データに対するニューロン・カバレッジ、縦軸に同じ入力データの予測確率をプロットした散布図である。赤い点は予測が正しいデータ、青い点は不正解のデータを表す。なお、訓練データ正解率と試験データ正解率は共に 99%であった。



(a) ニューロン・カバレッジ vs データ度数　(b) ニューロン・カバレッジ vs 予測確率

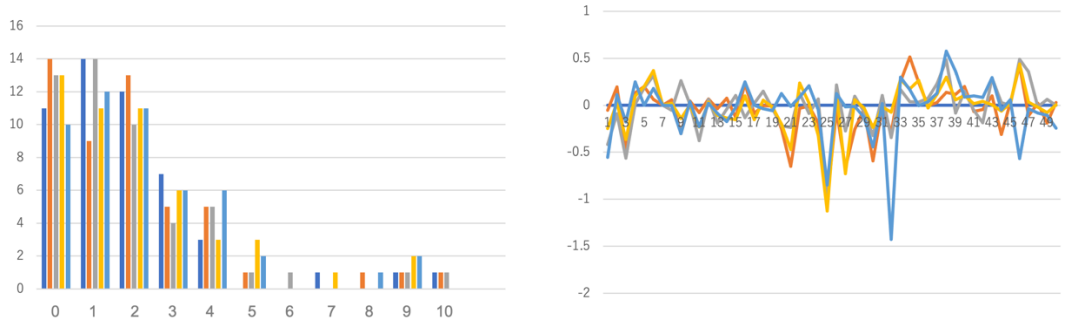
図 6.2　訓練データの分析結果

図 6.2 (a)は FCN 層でのニューロン・カバレッジが、0.38 から 0.84 の間に分布し、中央値ならびに平均値は 0.60 を示す。図 6.2 (b)はニューロン・カバレッジと予測分類確率の相関がほとんどないことを、あらためて確認する結果となった。訓練データによってニューロン・カバレッジに大きな違いがある一方、正解か不正解かに関わらず予測分類への寄与の大きさが異なると考えることができる。つまり、不正解であることに大きく寄与する場合もニューロンカバレッジが大きな値をとる。

図 6.2 中、たとえば、楕円で囲んだ領域は、平均よりも大きなニューロン・カバレッジとなる訓練データを表し、出力に有意な影響を与えた結果として、データごとに測定したような予測分類確率を導いた。したがって、対象データの特徴（分類確率）を忠実に表すと解釈できる。仮に、ニューロン・カバレッジが平均よりも小さい訓練データを選んだとする。図 6.2 (b)によると、出力の予測分類確率は、図 6.2 の楕円領域と同様に幅広くばらつく一方、ニューロン・カバレッジが小さいことから、出力に適切な影響を与えたか確かでない。つまり、M^S を得る際の訓練データとしての有用な役割が小さいと解釈できる。本章では、予測分類確率によらず小さいニューロン・カバレッジとなる訓練データを外れ値と考える。

次に、ニューロン・カバレッジをもとに訓練データを仕分けし、同じ大きさの訓練データ $S^{(K)}$ を作成する。次いで、$S^{(K)}$ を用いて訓練済み学習モデル $M^{(K)}$ を導出する。最後に共通の試験データを用いて $M^{(K)}$ を評価する。この実験では、大まかに訓練データを作成する訓練データ・デバッグ作業を想定した。つまり、仕分けした訓練データ全体を対象とし、デー

タ個々のレベルでの調整は行わなかった。



(a) 出力値の中央値 vs 試験データ度数　　(b) ニューロン vs ズレの大きさ

図 6.3 ニューロン活性ベクトル

図 6.3 は 5 つの訓練済み学習モデル $M^{(K)}$ の試験データによる評価結果を FCN 層のニューロン活性ベクトルによって表した。ここで、ニューロン活性ベクトルは内部特徴量であって、構成ニューロンの出力値を成分とする多次元ベクトルである。

得た訓練データセット $S^{(K)}$ は、ランダム選択・右側編集（図 6.2 の楕円内の訓練データを置き換え）・左側編集・両側編集・ランダム選択した後にデータ補完である。以上は図 6.3 (a)の棒グラフに左から順に対応する。試験データ正解率は 97.14%・96.89%・96.90%・96.94%・95.81%であり、$M^{(5)}$ を除いて大きな違いはない。$M^{(1)}$ から $M^{(4)}$ に違いがないことは訓練データの仕分け方法（図 6.2 (b)参照）と整合している。また、$S^{(5)}$ はデータ補完を施しているので試験データの分布特徴が異なることと整合する。

図 6.3 (a)はベクトル成分の値からなる分布を対象とし、その中央値が横軸の値の範囲内となった試験データの度数を表す。試験データの多くは中央値が小さく予測分類への寄与が小さいことを表す。また、$S^{(K)}$ によって分布が異なる。ここで、$S^{(1)}$ はランダム選択した訓練データセットであり、もととなった D の分布にしたがうとして良いだろう．図 6.3 (b)では、この $S^{(1)}$ を基準に考え、ニューロン活性ベクトルの各成分に対して、$S^{(K)}$ $(K = 2, \ldots, 5)$ の場合と $S^{(1)}$ の場合との差を示す。差が大きいほど $S^{(1)}$ の分布（したがって D の分布）による効果からのズレが大きい。つまり、訓練データ編集の効果が大きいことを示唆する。

以上から、ニューロン・カバレッジをもとにすることで、モデル正確性への影響が微小である一方で、ニューロン活性ベクトルのズレ（図 6.3 (b)）を生じるような方法での訓練データ加工が行うことができたと言える。$S^{(3)}$ は左側編集した結果であり、ニューロン・カバレッジが小さい訓練データを積極的に除去したものである。図 6.3 から定性的に判断する限り、$S^{(3)}$ が外れ値の除去に貢献していると考えられる。定量的な結論を導くには、モデル・ロバスト性の経験的な検査方法の確立が必要である。

6.5 おわりに

深層 NN ソフトウェアに不具合がある時、その直接原因は訓練済み学習モデルの欠陥であるが、このモデル構築に関わった情報に根本原因がある。2021 年度は訓練データの偏りが不具合の根本原因となる場合を対象とし、訓練データのデバッグ・テスティング法を検討し

た。ここで、不具合はモデル正確性とモデル・ロバスト性の 2 つの品質観点から判断するものである。一般にこれら 2 つの側面がトレードオフ関係にあることを考慮して訓練データをデバッグする必要がある。そこで、訓練データ・デバッグ方法の要点を、プライバシー品質の文脈で論じられてきた帰属関係推定法の知見をもとに整理し、外れ値検知の問題に帰着した。しかし、どのように外れ値を定義するかは自明でない。モデル内部の活性状態からニューロン・カバレッジを計算し、その偏りから訓練データの外れ値を推定する方法を提案した。実験によって、訓練データ・デバッグを補助する情報が得られることがわかった。

今後、実験を精密化し系統的に行うことで、訓練データ・デバッグを目的とする外れ値の検出法を確立する。さらに、モデル正確性とモデル・ロバスト性の 2 つの観点から、デバッグを進める方法を整理する。

7 ロバストネスの評価・向上技術

本章におけるロバストネスは、入力ノイズ（敵対的データも含む）に対する機械学習モデルの耐性である。例えば、どの程度ノイズを付加しても推論結果を維持できるかを評価する。そのロバストネスの指標の一つに最大安全半径がある。本章では、順伝播型ニューラルネットワークを用いた分類器を対象として、敵対的データと最大安全半径について説明した後、最大安全半径を計測する技術と増加させる技術についての調査結果について報告する。

7.1 ロバストネスの指標（最大安全半径）

機械学習ソフトウェアにおいては、訓練済み学習モデル（正確には、学習モデルにもとづく推論プログラム）は、わずかにノイズを付加された入力データによっても誤推論する問題が知られている。そのような誤推論させる入力データは敵対的データ（adversarial example）[54]と呼ばれており、近年、敵対的データに対する研究が活発に行われている。入力データ$x \in \mathbb{R}^n$（$\mathbb{R}$は実数の集合）のδ近傍（半径$\delta \in \mathbb{R}$の球の内側）に含まれる全ての敵対的データの集合$Adv_\delta(x)$は次のように定義できる。

$$Adv_\delta(x) = \{x' \mid \| x - x' \| \leq \delta \ \wedge \ f(x) \neq f(x')\}$$

ここで、$f(x)$は入力xに対する機械学習モデルの出力（分類結果）を表す関数、$\|x - x'\|$は 2 つのデータx, x'の距離（差）を表す。距離の定義にはpノルムが使われることが多い。

図 7.1 を用いて敵対的データについて説明する。図 7.1 の左側はニューラルネットワークへの入力空間、右側がニューラルネットワークからの出力空間を表す。入力空間の赤い球の中心がパンダ画像（元データ）であり、その球（半径δ）の内側が大きさδ未満の微小ノイズを付加した画像の集合（δ近傍）である。このδ近傍の入力画像の集合に対するニューラルネットワークの出力の集合が、右側の出力空間の赤い領域である。ここで、出力側の赤い領域の右下の決定境界を超えて猿の領域に入っている部分が誤分類を表しており、この部分にマップされる入力が敵対的データである。

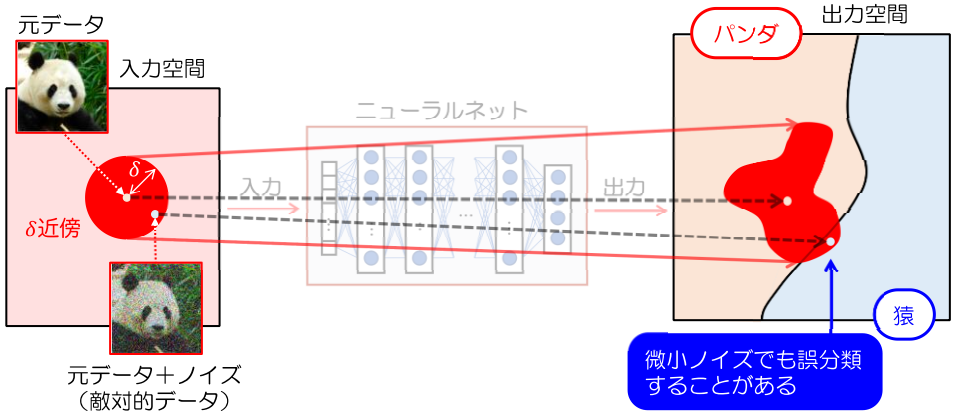


図 7.1 敵対的データ（ノイズ付加パンダ画像を猿に誤分類）

入力データxのδ近傍（xを中心とする半径δの球の内側）に敵対的データが存在しないと

き、δをxの安全半径という。特にxの安全半径の中で最大の半径（最大安全半径, Maximum Safe Radius）$MSR(x)$は次のように定義される。

$$MSR(x) = \max\{\delta \mid Adv_\delta(x) = \emptyset\}$$

この最大安全半径が大きいほど、敵対的データによる攻撃は難しくなるため、機械学習モデルのロバストネス（敵対的データを含む入力ノイズに対する耐性）の指標のひとつとして最大安全半径を使うことができる。

図 7.1 のδ近傍の入力画像の一部は猿に誤分類されるため、このδは安全半径ではない。一方、図 7.2 のδ近傍の入力画像は誤分類されることはないため、このδは安全半径であり、これ以上δを大きくすると誤分類される可能性がある（敵対的データを含む）ため、図 7.2 のδは最大安全半径である。

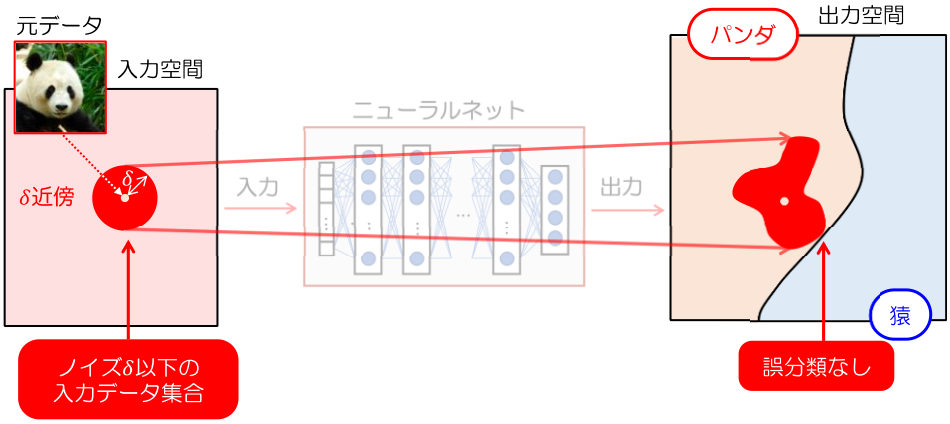


図 7.2　最大安全半径 δ

7.2　ロバストネスの評価・向上技術調査結果

ロバストネスの評価と向上に関する研究論文について調査した結果を表 7.1 に示す。ロバストネスについては多くの研究論文が発表されているが、比較的良い結果が得られている最近の論文を代表として示している。表 7.1 の各論文の下には、その論文で提案されている手法を適用できるスケールの参考情報として、その手法の評価実験に使用されたニューラルネットワークの情報を記入している。表 7.1 は次の観点から整理している。

- 横方向（技術の用途）：
  - ロバストネスの評価：最大安全半径の見積り
  - ロバストネスの向上：指定された最大安全半径をもつデータ数の増加
- 縦方向（評価の信頼度・精度）：
  - 保証あり：δ近傍に敵対的データが存在しないことを保証できる
    - 厳密：最大安全半径の厳密な見積り
    - 近似：最大安全半径よりも小さめ（安全半径のひとつ）の見積り
      - 決定的：δ近傍に敵対的データは存在しない（100%安全）
      - 確率的：δ近傍に敵対的データが存在しない確率はp%
  - 保証なし：δ近傍に敵対的データが存在しないことを保証できない

表 7.1　ロバストネス（最大安全半径）の評価と向上に関する技術

<table>
<tr><th colspan="3"></th><th>ロバストネスの評価</th><th>ロバストネスの向上</th></tr>
<tr><td rowspan="4">保証あり</td><td rowspan="2">厳密</td><td rowspan="2"></td><td>最大安全半径の厳密な見積り
Katz et al. 2017 (Reluplex) [55]
ACAS-XU-DNN, 300 ReLU nodes
6 hidden layers,
数百ノード程度が上限

Tjeng et al. 2019 [56]
CIFAR-10, ResNet, 9-CNN, 2-layer,
107,496 ReLU units,
Reluplex より 100～1,000 倍程度高速</td><td></td></tr>
<tr><td rowspan="3">近似</td><td>決定的</td><td>最大安全半径の小さめの見積り
Weng et al. 2018 (Fast-Lin) [57]
CIFAR, 6-layer, 12,288 ReLU units
Reluplex より 10,000 倍程度高速

Boopathy et al. 2019 (CNN-Cert)[58]
CIFAR-10 (32x32x3), 5-layer,
10 filters,　29,360 hidden nodes,
Fast-Lin より高速</td><td>近傍に敵対的データがないように訓練
Wong and Kolter 2018 [62]
SVHN (32x32x3), 2-conv, 32-ch,
100, 10 hidden units, ReLU,
ImageNet への適用は困難</td></tr>
<tr><td>確率的</td><td>確率的最大安全半径の小さめの見積り
Weng et al. 2019 (PROVEN) [59]
CIFAR, 5-layer, CNN, ReLU
CNN-Cert と同程度</td><td>訓練後に保証付ランダムスムージング
Lecuyer at el. 2019 [63]
ImageNet (299x299x3),
Inception-v3 + auto-encoder

Cohen at el. 2019 [64]
ImageNet (299x299x3),
ResNet-50 (50-layer)
Lecuyer [63]より精度向上</td></tr>
<tr><td>保証なし</td><td colspan="2"></td><td>最大安全半径の大きめの見積り
Carlini and Wagner 2017 [60]
ImageNet (299x299x3),
Inception-v3

最大安全半径のおおよその見積り
Weng et al. 2018 (CLEVER) [61]
ImageNet (299x299x3),
ResNet-50 (50-layer)</td><td>近傍の敵対的データを探索しながら訓練
Madry et al. 2018 [65]
CIFAR (32x32x3),
28-10 wide ResNet</td></tr>
</table>

以降、小節 7.2.1～7.2.7 で表 7.1 の各手法について簡単に説明する。

7.2.1　ロバストネス評価、保証あり（厳密）

Katz 等は、与えられた性質を機械学習モデルが満たすことを判定する方法 Reluplex を提案した[55]。その方法を実装した実証用ツールも公開されている。性質は入出力関係につい

ての制約であり、入力データの δ 近傍に敵対的データが存在しないことを網羅的に厳密に（健全かつ完全に）判定できる。すなわち、二分探索などで半径 δ を変えながら敵対的データの有無を判定することによって、最大安全半径を見積もることができる。Reluplex は Simplex 法（線形計画問題の解法のひとつ）に ReLU 関数用の規則を追加した解法であり、実数を扱える充足可能性判定ツール（SMT-Solver）によって実装されている。ロバストネス以外の性質も判定できる強力なツールであるが、計算コストが高く、扱えるニューロン数が ReLU 数百個程度という制約がある。

Tjeng 等は、最大安全半径を効率よく計算する方法を提案し、その方法を混合整数線形計画法ソルバ（MILP）上に実装して、10 万個の ReLU 型ニューロンをもつネットワークの最大安全半径を厳密に求められることを示した[56]。まだ実用的な機械学習モデルに適用するには十分とは言えないが、このようにスケーラビリティ改善の研究は進んでいる。

7.2.2 ロバストネス評価、保証あり（近似、決定的）

Weng 等は、ReLU 型ニューラルネットの最大安全半径を近似的に見積る方法 Fast-Lin を提案した[57]。Fast-Lin では、図 7.3 に示すように、出力領域を多角形で線形近似し、最大安全半径よりも少し小さめの近似値δを見積もる。この近似値δは最大安全半径を超えないため（健全）、δ近傍に敵対的データが存在しないことを保証できる。すなわち、δは安全半径のひとつであり、最大安全半径の下界である（$\delta \leq MSR(x)$）。多角形で線型近似することによって、厳密な方法（Reluplex）よりも 1 万倍以上速い結果が得られたことが報告されている。

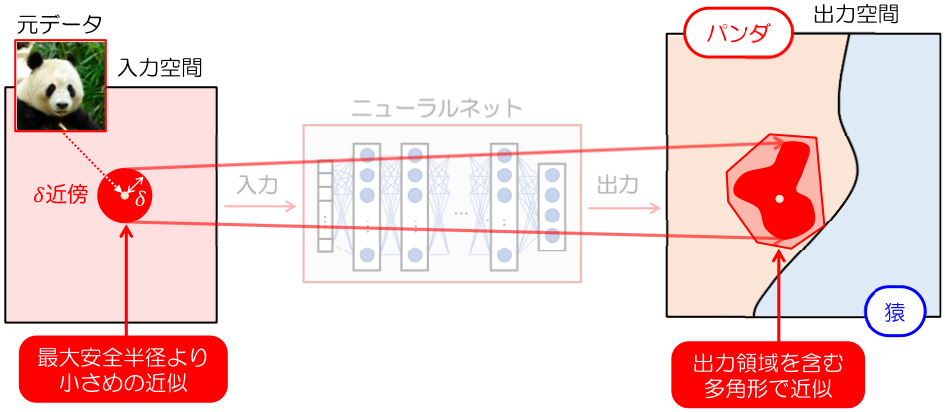


図 7.3 最大安全半径の近似値（少し小さめ）の見積り

Boopathy 等は Fast-Lin を改良した手法 CNN-Cert を提案した[58]。CNN-Cert では、ReLU 以外の活性化関数（sigmoid, tanh, arctan）を含む Convolutional ネットワークにも対応し、Fast-Lin よりも近似精度と計算速度を向上した。

7.2.3 ロバストネス評価、保証あり（近似、確率的）

Weng 等は、確率的な最大安全半径を近似的に見積もる方法 PROVEN を提案した[59]。安全確率ρの最大安全半径δは、図 7.4 に示すように、δ近傍に敵対的データが存在しない確率

がρである、すなわち、$(1-\rho)$の確率で敵対的データが存在することを許容する最大の半径である（決定的な最大安全半径は安全確率1の最大安全半径に相当する）。PROVEN は、CNN-Cert をベースに開発されおり、計算量は CNN-Cert から大きくは増えていない。

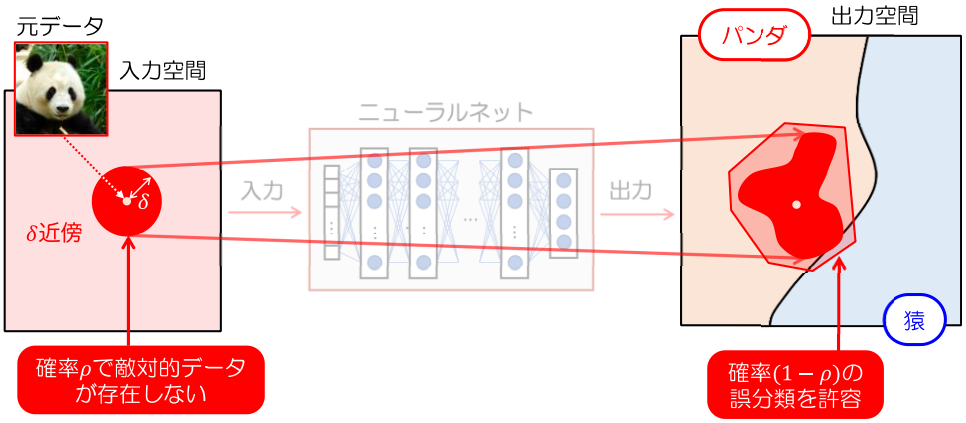


図 7.4　安全確率ρの最大安全半径の近似値の見積り

7.2.4　ロバストネス評価、保証なし

Carlini と Wagner は、既存の最適化ツール（Adam）を用いて、入力データxに最も近い敵対的データを探索し、その距離δを見積もる方法を提案した[60]。ただし、この方法で得られる距離δが実際に敵対的データまでの最短距離である保証はなく、その距離よりも近いところに敵対的データが存在する可能性はある。すなわち、最大安全半径の上界である（$msr(x) \leq \delta$）。この方法で得られる距離δが安全半径である保証はないが、最大安全半径の目安として、最近のロバストネスの論文ではしばしば評価に使われている。

Weng 等は、攻撃方法に依存しないロバストネスの評価値として、おおよその最大安全半径を求める方法 CLEVER を提案した[61]。比較的大きなネットワークにも適用可能な方法であり、画像認識モデル Inception-v3 を 10 秒程度で評価できたと報告している。入力のわずかな変化が出力に与える影響の最大値を極値理論によって推定し、最大安全半径に近い値δを見積もっている。図 7.5 に示すように、見積もった値δは実際の最大安全半径より大きいこともあるため、δ近傍内に敵対的データが存在する可能性はある（安全半径である保証はない）。

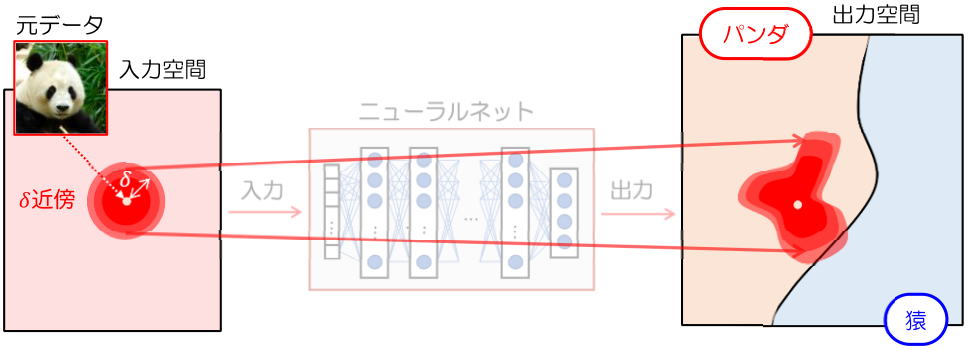


図 7.5　最大安全半径の近似値の見積り（保証なし）

7.2.5 ロバストネス向上、保証あり（近似、決定的）

Wong 等は、訓練データセットの各データの最大安全半径がδ（指定値）になるように訓練する方法（ロバスト訓練）を提案した[62]。この方法は訓練後に全ての訓練データに対して最大安全半径δを保証するものではないが、各入力データの最大安全半径の近似値（安全半径）を見積もる方法を与えている。このロバスト訓練では、各訓練データのδ近傍で最も誤推論する可能性のあるデータに対して正しい推論をするように訓練する。

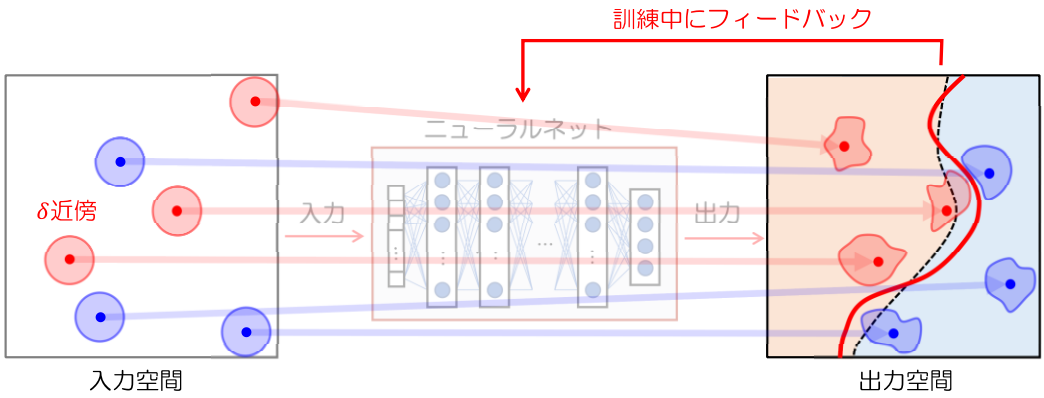


図 7.6 ロバスト訓練（入力のδ近傍も含めた訓練）

ロバスト訓練の様子を図 7.6 に示す。図 7.6 の出力空間の点線は通常の訓練によって学習した決定境界、赤色線はロバスト訓練によって学習した決定境界を表す。入力空間の 6 個の訓練データについては両方の境界線によって正しく分類されているが、各訓練データのδ近傍のデータについては点線の境界線（通常訓練）では誤分類が生じている。一方、ロバスト訓練では赤色の境界線のように、δ近傍のデータについても正しい分類になるように訓練を行う。Wong 等のロバスト訓練は、ロバストな学習モデルを保証付で訓練する方法であるが、スケーラビリティが低く、訓練可能なネットーワークのサイズを大きくできない問題がある。この論文は、MNIST（28×28）と SVHN（32×32）のデータセットに対して有効性を示しているが、ImageNet（256×256）には適用できなかったと報告している。

7.2.6 ロバストネス向上、保証あり（近似、確率的）

Lecuyer 等はランダムスムージングによって確率的に保証可能な最大安全半径を見積もる方法を提案した[63]。ランダムスムージングとは、入力データに防御用のノイズを付加した推論を繰り返し、その複数の出力の平均値を最終出力とする方法である。

ランダムスムージングの様子を図 7.7 に示す。図 7.7 の出力空間の点線は防御用ノイズを付加しない場合の決定境界、赤色線は防御用ノイズを付加した場合の決定境界を表す。ランダムスムージングは決定境界を滑らかにすることによってロバストネスを向上させる技術であり、ImageNet（299×299×3）のような大きな入力データに対する機械学習モデルのロバストネスの保証にも成功している。付加するノイズの分散を増加させると保証可能な最大安全半径も増加するが、一方で正解率等の推論精度は低下する。この論文では、差分プライバシの技術（類似した 2 つの入力に対する出力をノイズ等によって統計的に区別でき

なくする技術）を適用し、保証可能な最大安全半径、防御ノイズ付推論回数、許容される敵対的データの存在確率等の関係を明確にした。

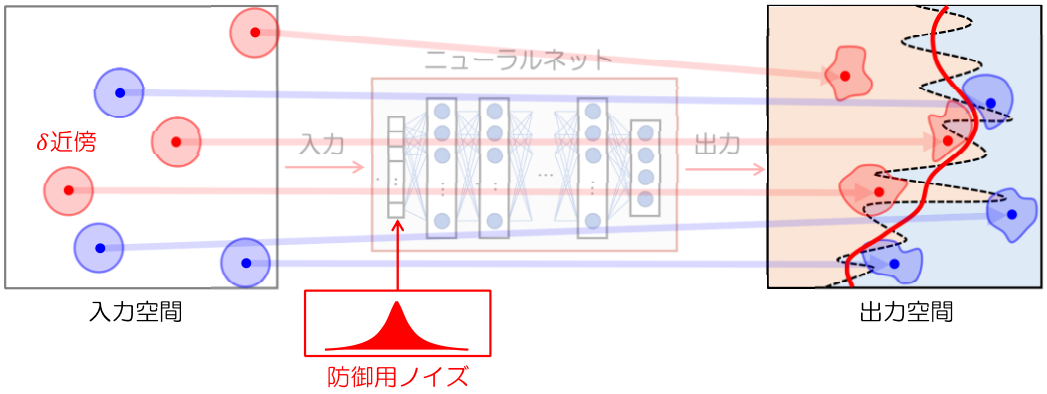


図 7.7　ランダムスムージングによるロバストネスの向上

Cohen 等は、ランダムスムージングによって確率的に保証可能な最大安全半径を、Lecuyer 等の方法[63]よりも精度良く（より大きく）見積もる方法を提案した[64]。

ランダムスムージングは 1 回の推論を行うために内部で複数回（実験的には数十～数百回程度）推論する必要はあるが、Lecuyer や Cohen 等の研究によって、大規模なネットワークに対しても、ロバストネスを確率的に保証することが可能になる。

7.2.7　ロバストネス向上、保証なし

Madry 等は訓練データセットの各データの最大安全半径がδ（指定値）になるように訓練する方法（敵対的訓練）を提案した[65]。この方法では、訓練中に各訓練データのδ近傍で敵対的データになる可能性のあるデータを探索し、そのデータに対しても正しい推論を行えるように訓練する。Wong 等の保証付のロバスト訓練[62]と比較して、ロバストネスを保証することはできないが、ロバスト訓練よりも大きなネットワークに適用可能である。ランダムスムージングのように推論時に複数回の推論を行う必要がなく、ロバストネス向上のための候補技術になりうる。

7.3　まとめ

一般に、ロバストネスを向上させると正解率が低下する傾向にあり、現在は正解率などの評価指標の方が重視されることが多い。しかし、ロバストネスを考慮しない場合、わずかなノイズでも正解率が低下する可能性があるため、誤判断によるリスクが高い場合はロバストネスによる評価も重要である。今回調査したロバストネスに関する技術は本報告書執筆時（2019 年頃）の論文で提案されたものであり、まだこれらの技術を容易に利用できる評価環境は整備されていないが、技術的には実用的なニューラルネットワークにも適用可能になりつつある。今後、そのような評価環境が整備されれば、ロバストネスもニューラルネットワークの一般的な評価指標のひとつになりうると考える。

8　汎化誤差上界の見積り技術

本章では、未知の入力データに対する機械学習の振舞い保証を目的として、順伝播型ニューラルネットワーク、特に分類器の汎化誤差上界の見積法の調査と実験結果について報告する。汎化誤差とは全ての入力に対する分類器の出力の不正解率の期待値である。

8.1　汎化誤差と経験誤差

本報告書では、図 8.1 に示すような機械学習を用いたニューラルネットワークを、その入出力関係でモデル化する。特に、教師ありの深層学習によって分類器として訓練した順伝播型ニューラルネットワークの入出力関係を対象とし、入力xとその出力（推論結果）yとの関係を関数 $y = h_w(x)$ によって表記する。この関数h_wを**仮説**と呼ぶ。ここで、w はニューラルネットワークのニューロン間の結合の重み（訓練パラメータ）である。重みの候補集合を$\mathcal{W}$とするとき、教師ありの訓練とは、訓練データセットに適合する重みを候補集合$\mathcal{W}$から選択することである。このとき、仮説候補の集合は $\mathcal{H} = \{h_w \mid w \in \mathcal{W}\}$ である。この集合$\mathcal{H}$は訓練対象のニューラルネットワークが表現可能な関数の集合であり、この集合を**仮説集合**と呼ぶ。なお、重みが重要でない場合は w を省略して$y = h(x)$と書くこともある。

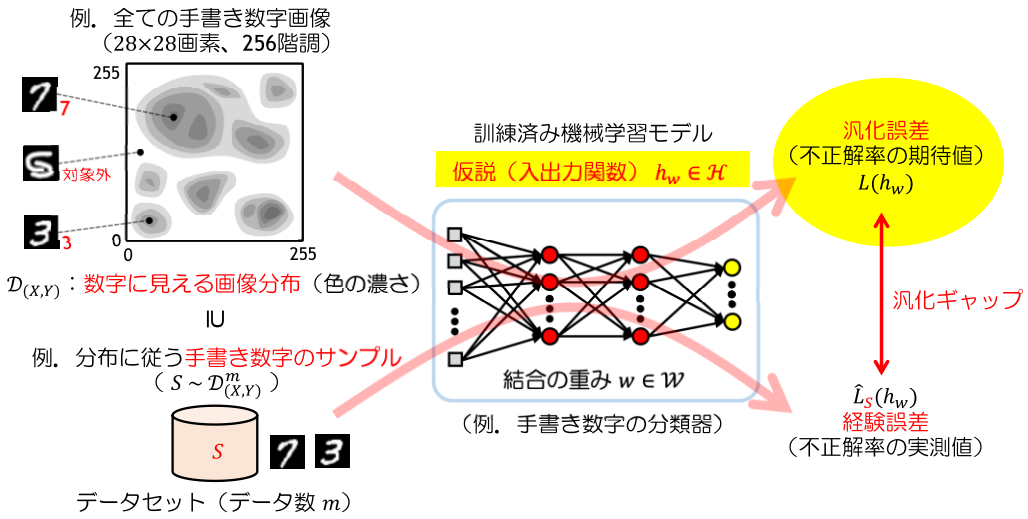


図 8.1　汎化誤差と経験誤差

このとき、**汎化誤差**とは、分布$\mathcal{D}$にしたがう全入力データに対する分類器（仮説 h_w ）の不正解率の期待値 $L(h_w)$ であり、次式によって定義される。

$$L(h_w) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbb{I}(y \neq h_w(x))]$$

ここで、$\mathbb{I}(b)$ は次のように定義される指示関数であり、上の式の場合は不正解ならば 1（この値は損失値と呼ばれる）、正解ならば 0 を返す関数として使われている。

$$\mathbb{I}(b) = \begin{cases} 1 & (b = \text{true}) \\ 0 & (b = \text{false}) \end{cases}$$

例えば、図 8.1 左上の入力空間（簡単のため2次元で図示しているが、実際には784次元）内の色の濃さは「数字に見える画像（$28 \times 28 = 784$画素、256 階調）の分布$\mathcal{D}$」を表しており、その数字に見える全ての画像を入力したときの不正解率の期待値が汎化誤差である（空間内の全画像（256^{784}個）の不正解率ではないことに注意）。

一方、**経験誤差**とは、分布$\mathcal{D}$にしたがうm個の入力データサンプルの集合 $S \sim \mathcal{D}^m$ に対する仮説 h_w の不正解率 $\hat{L}_S(h_w)$ であり、次式によって定義される。

$$\hat{L}_S(f_w) = \frac{1}{m} \sum_{(x,y) \in S} [\mathbb{I}(y \neq h_w(x))]$$

特に、訓練用のデータセットSに対する経験誤差 $\hat{L}_S(h_w)$を**訓練誤差**、テスト用のデータセットTに対する経験誤差 $\hat{L}_T(h_w)$を**テスト誤差**と呼ぶ。

8.2 汎化誤差上界見積法の解説

一般に入力空間には無数の入力データが存在するため、汎化誤差を正確に計算することは困難であるが、これまでに様々な汎化誤差上界見積法（仮説 h の汎化誤差が○%以下であることを確率的に保証する方法）が提案されている。Valle-Pérez と Louis は、そのような汎化誤差上界見積法を適用条件によって分類し、論文[66]の表 1 に示した。表 8.1 に、論文[66]の表 1 を簡略化し、RATT 方式[76]を追加した汎化誤差上界見積法の分類表を示す。本節では表 8.1 の各汎化誤差上界見積法について簡単に説明する。

表 8.1 汎化誤差上界見積法の分類

	訓練アルゴリズム独立	訓練アルゴリズム依存
訓練データ独立	VC 次元方式 仮説集合の複雑度を用いる	圧縮方式 訓練後の仮説の訓練データセットへの依存度を用いる
訓練データ依存	ラデマッハ複雑度方式 仮説集合とデータセットの複雑度を用いる	マージン方式 出力マージン付を考慮した訓練誤差を用いる 安定性方式 確率的訓練アルゴリズムの安定性を用いる RATT 方式 ランダムラベルデータに対する訓練誤差を用いる 感度方式 重みにノイズを付加した場合の訓練誤差を用いる 周辺尤度方式 仮説集合の仮説分布の周辺尤度を用いる

8.2.1 VC 次元方式

VC 次元（Vapnik-Chervonenkis） $VC(\mathcal{H})$ は仮説集合 $\mathcal{H}$ の複雑度であり、$\mathcal{H}$によって二分可能なデータ数の最大値のことである[67]。汎化誤差上界は、仮説の候補として用意した仮

説集合 $\mathcal{H}$ の VC 次元を用いて次の不等式によって見積もることができる。この不等式は、任意の訓練データセット$S\sim\mathcal{D}^m$（サイズm）、任意の仮説$h \in \mathcal{H}$について、確率 $(1-\delta)$ 以上で成り立つことが示されている[66]。

$$L(h) \leq \hat{L}_S(h) + 144\sqrt{\frac{VC(\mathcal{H})}{m}} + \sqrt{\frac{\ln\frac{1}{\delta}}{m}}$$

VC 次元による汎化誤差上界見積りでは、仮説集合$\mathcal{H}$が与えられたとき、任意のデータセットに対して任意の仮説が選択された場合の汎化誤差上界の最悪値を見積もるため、その値は 100% 大きく超えることが多く、訓練済み学習モデルの汎化性能の評価に利用することは難しい。

8.2.2 ラデマッハ複雑度方式

ラデマッハ複雑度（Rademacher complexity）は、データセットに対する仮説集合の複雑度である[67]。汎化誤差上界は、与えられた仮説集合 $\mathcal{H}$ と訓練データセット $S\sim\mathcal{D}^m$に対するラデマッハ複雑度 $R(S,\mathcal{H})$ を用いて、次の不等式によって見積もることができる。この不等式は、任意の仮説$h \in \mathcal{H}$について、確率 $(1-\delta)$ 以上で成り立つことが示されている[66]。

$$L(h) \leq \hat{L}_S(h) + 2R(S,\mathcal{H}) + 4c\sqrt{\frac{2\ln\frac{4}{\delta}}{m}}$$

ここで、cは定数である。訓練データセットを考慮しているため、VC 次元方式より精度は向上できるが、任意の仮説が選択された場合の汎化誤差上界の最悪値を見積もるため、汎化誤差上界は 100%を超えることは多く、学習モデルの汎化性能の評価に利用することは難しい。

8.2.3 マージン方式

出力マージンとは、出力層の正解クラスのニューロンの出力値とそれ以外のニューロンの出力値の最大値との差である。例えば、図 8.2 のような出力層の各ニューロンの出力値の例の出力マージンは0.1である。2 つの仮説が、あるデータセット に対して同じ経験誤差（不正解率）をもつ場合でも、出力マージンの大きい仮説の方がノイズなどに対して耐性があり、一般に安定した推論を行うことができる。そのような出力マージンを考慮した経験誤差は次式により定義される。

$$\hat{L}_{S,\gamma}(h) = \frac{1}{m}\sum_{(x,y)\in S} l_\gamma\big(h,(x,y)\big),$$

ここで、$l_\gamma\big(h,(x,y)\big)$ は次式により定義されるマージン付損失関数である。

$$l_\gamma\big(h,(x,y)\big) = \mathbb{I}\left(h(x)[y] \leq \gamma + \max_{y'\neq y} h(x)[y']\right)$$

ここで、$\mathbb{I}$は 8.1 節の指示関数、$h(x)[y]$はニューラルネットワークの出力層におけるクラス y の出力ニューロンの値である（ $h(x) = \mathrm{argmax}_y(h(x)[y])$ ）。マージン付経験誤差 $\hat{L}_{S,\gamma}(h)$ では、正解クラスのニューロンの出力値が他のニューロンの出力値の最大値より大きくても、その差（出力マージン）が閾値 γ 以下ならば不正解としてカウントされる。

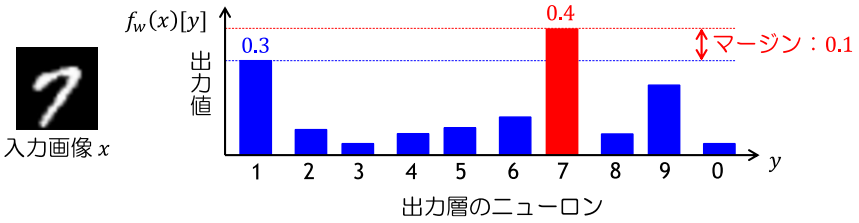


図 8.2 手書き数字分類器の出力マージンの例

マージン方式では、訓練データセット $S \sim \mathcal{D}^m$ で訓練した重み w（この w により定まる仮説を h_w と書く）とマージン閾値 γ を用いて、仮説 h_w の汎化誤差上界を次の不等式によって見積もることができる。この不等式は確率 $(1-\delta)$ で成り立つことが示されている[68][69]。

$$L(h_w) \leq \hat{L}_{S,\gamma}(h_w) + \sqrt{\frac{\left(42\sum_{i=1}^{d}\left(2\sqrt{\omega_i}+\sqrt{2\ln(2d)}\right)\right)^2 \prod_{i=1}^{d}\|w_i\|_2^2 \times \sum_{i=1}^{d}\frac{\|w_i\|_F^2}{\|w_i\|_2^2} + \ln\left(\frac{m}{\delta}\right)}{\gamma^2 m}}$$

ここで、d は層数、w_i は第 i 層への重み行列、ω_i は w_i の要素数である。なお、$\|w_i\|_2$ は行列 w_i のスペクトルノルム、$\|w_i\|_F$ は w_i のフロベニウスノルムである。

この不等式の右辺の第 2 項は出力マージンの閾値 γ を大きくすると汎化ギャップ（汎化誤差と経験誤差の差）が小さくなることを表している。一方で、右辺の第 1 項のマージン付訓練誤差 $\hat{L}_{S,\gamma}(h)$ は γ とともに増加する。これらのことは、出力マージンの大きい仮説の汎化誤差は小さくなる（汎化性能は高くなる）ことを意味しており、経験的な傾向とも一致する。一般に、この不等式による上界の見積りも 100% を超えることは多く、汎化性能の絶対的な評価への適用は難しいが、閾値 γ は汎化性能を相対的に評価する指標（汎化尺度）として有効である[68]。

8.2.4 感度方式

感度方式は PAC-Bayesian と呼ばれる分析方法を基礎にしている。PAC-Bayesian では、訓練データセットから得られる情報量を、訓練前と後のパラメータの分布の違い（KL 情報量と呼ばれる）から見積もっており、この情報量が小さいほど訓練データセットへの依存性が小さくなる（汎化誤差が小さくなる）ことを利用している。

感度方式では、訓練前後の重みの分布の違いを用いる。訓練前の各重み（要素数 ω）を正規分布 $\mathcal{N}(0,\sigma^2)$（平均 0、標準偏差 σ）の乱数で初期化し、訓練データセット S で訓練後の各重みに正規分布 $\mathcal{N}(0,\sigma^2)$ のノイズを付加した場合の汎化誤差の期待値の上界は、その訓練誤差の期待値とノイズの大きさ σ を用いて、次の不等式によって見積もられる。この不等式は確率 $(1-\delta)$ で成り立つことが示されている[66]。

$$\mathbb{E}_{u\sim\mathcal{N}(0,\sigma^2)^\omega}[L(h_{w+u})] \leq \mathbb{E}_{u\sim\mathcal{N}(0,\sigma^2)^\omega}\left[\hat{L}_S(h_{w+u})\right] + 4\sqrt{\frac{1}{m}\left(\frac{\|w\|_2^2}{2\sigma^2} + \ln\frac{2m}{\delta}\right)}$$

この不等式の右辺の第 1 項 $\mathbb{E}_{u\sim\mathcal{N}(0,\sigma^2)^\omega}\left[\hat{L}_S(h_{w+u})\right]$ は、訓練済みの重みに正規分布ノイズを付加した場合の訓練誤差 $\hat{L}(h_{w+u},S)$ を複数回測定した平均値によって近似できる。

この不等式の右辺の第 2 項はノイズの標準偏差 σ を大きくすることによって汎化ギャッ

プを小さくできることを示している。一方で、右辺の第 1 項はノイズ σ とともに増加する。これらのことは、ノイズに対して耐性のある（感度の低い）仮説の汎化誤差は小さくなることを意味しており、経験的な傾向とも一致している。一般に、この感度による汎化誤差上界見積法でも、8.2.3 小節のマージン方式による見積法と同様に 100% を超えることが多いため、絶対的な汎化性能評価は難しい。一方、マージン方式の閾値 γ と同様に、σ は汎化性能を相対的に評価する指標（汎化尺度）として有効である[68]。

8.2.5 圧縮方式

圧縮方式を適用するためには、訓練データセット Sで訓練した仮説 h について、次の二つの条件を満たすように、S を依存部Tと独立部V に分割する必要がある：

- 仮説 h は依存部 T のみに依存する（$h = B(T)$となるような写像 B が存在する）こと
- 独立部 V に対する経験誤差をゼロ（$\hat{L}_V(h) = 0$）にすること

訓練データセット S（m個）を依存部 T（k個）と独立部 V（$m-k$個）に分割できれば、仮説hの汎化誤差上界を次の不等式によって見積もることができる。この不等式は、確率 $(1-\delta)$ 以上で成り立つことが示されている[66]。

$$L(h) \leq \frac{8k}{m} \ln \frac{m}{\delta}$$

圧縮方式では、データセットへの依存度 (k/m) を下げることによって、汎化誤差上界の見積り精度を向上させる（汎化誤差上界を下げる）ことができる。一方、データセットを依存部と独立部に分割する方法（圧縮スキームと呼ばれる）については、Brutzkus 等[71]が、確率的勾配降下法で訓練した 2 層ニューラルネットワークのための方法を開発したが、より汎用的なネットワークについてはまだ研究段階にある。

8.2.6 安定性方式

安定性方式では、確率的アルゴリズムの安定性をもとに汎化誤差上界を見積もる方法である。確率的アルゴリズム $\mathcal{A}$ の安定性は、訓練データセット S のなかの任意の一つデータサンプルを別のサンプルに置き換えたときの損失の増分によって定義される。その損失の増分が小さいほど確率的アルゴリズムは安定している（訓練データセット S への依存性が低い）ことを意味しており、汎化誤差上界を低くすることができる。

確率的アルゴリズム $\mathcal{A}$ の安定性を利用して、サイズmのデータセット S からランダムに選択した k 個（ $k \leq m$ ）のサンプルのデータセット S_k で訓練したときの汎化誤差と経験誤差の差の上界は次の不等式によって見積もることができる[72]。

$$\mathbb{E}_{(S\sim\mathcal{D}^m,\ \mathcal{A})}\left[L\left(\mathcal{A}_{S_k}\right) - \hat{L}_S\left(\mathcal{A}_{S_k}\right)\right] \leq \mathcal{O}\left(\sqrt{c\left(L\left(h_{w_1}\right) - L^*\right)} \cdot \frac{\sqrt[4]{k}}{m} + c\sigma\frac{\sqrt{k}}{m}\right)$$

ここで、$\mathcal{A}_S$はデータセット S に適合するように確率的アルゴリズム $\mathcal{A}$ によって訓練して得られる仮説、c は確率的アルゴリズムによる 1 回あたりの重みの更新量に関する定数、h_{w_1} は初期の重み w_1で決まる初期の仮説、L^* は仮説集合 $\mathcal{H}$ で達成可能な最小誤差、σ は訓練中

の確率的な勾配の標準偏差である。

論文[72]では、汎化誤差に近い上界の見積り結果が得られているが、損失関数がスムーズであることや各データサンプルは高々1 回使用すること（one-pass）という制限がある。現在も、安定性方式による汎化誤差上界見積法の適用条件を緩和（例えば、non-smooth な損失関数や multi-pass に対応）するための研究が進められている。

8.2.7 周辺尤度方式

周辺尤度方式は、8.2.4 小節の感度方式と同様に PAC-Bayesian を基礎にしているが、重みの分布ではなく仮説（入出力関数）の分布を用いるところが異なっている。仮説 $h_w \in \mathcal{H}$ は重み $w \in \mathcal{W}$ によって決まる関数であるが、一般に複数の重み w, w' が同じ仮説 $h_w = h_{w'}$ に対応するため、重み集合 $\mathcal{W}$ 上よりも仮説集合 $\mathcal{H}$ 上で訓練前と後の分布を比較した方が分布の違いを絞ることができ、汎化誤差上界を下げることができる。ただし、周辺尤度方式を適用するためには、次の二つの条件を満たすように訓練する必要がある：

- 訓練後の仮説の経験誤差をゼロ（$L_S(h) = 0$）にすること
- 訓練アルゴリズムは経験誤差ゼロの仮説候補を均一にサンプルすること

これらの条件を満たすならば、仮説 h の訓練前と訓練後の分布を各々 $P(h)$ と $Q(h)$ とすると、経験誤差をゼロにする任意の仮説 $h \in \mathcal{H}_0(S) = \{h \in \mathcal{H} \mid \hat{L}_S(h) = 0\}$ について、

$$Q(h) = \frac{P(h)}{M_0(S)}, \quad where\ M_0(S) = \sum_{h \in \mathcal{H}_0(S)} P(h)$$

が成り立つ。ここで、$M_0(S)$ は事前分布 $P(h)$ において S に対する訓練誤差がゼロの仮説を選択する確率であり、S の周辺尤度に相当する。確率的勾配降下法 SGD はこの条件を満たす傾向にあることが示されている[73]。

上記の条件が満たされるとき、$Q(h) \geq 1 - \gamma$ となるように選択した仮説 h について、汎化誤差の上界は次の不等式によって見積もることができる。この不等式は確率 $(1 - \delta)$ 以上で成り立つことが示されている[66]。

$$L(h) < 1 - \left(\frac{M_0(S)\, \gamma\, \delta}{m} \right)^{\frac{1}{m-1}}$$

ニューラルネットワークでは重み w を仮説 h_w に写像するときに単純化バイアスが働くため[73]、重みを一様乱数で初期化した場合でも、その仮説分布 $P(h)$ は一様にはならず、単純な（コルモゴロフ複雑度の低い）仮説ほど確率密度が高い分布になる傾向がある。一般に、（例えば、人工的なランダムな関数ではなく）現実世界の仮説は規則性や構造をもつと推測されるため、現実世界を反映する仮説 h の確率密度 $P(h)$ は高くなり、$M_0(S)$ や γ を大きくとれると考えられる。このことは、上の不等式の右辺の第 2 項を大きくするため、汎化誤差上界が下がることを意味する。

論文[66]にて、いくつかの仮定のもとで周辺尤度方式の最適性は証明されているが、汎化誤差上界の見積り実験では 2 クラス分類問題に制限されており、まだ十分な比較評価実験は示されていない。マルチクラス分類問題への適用の他、周辺尤度 $M_0(S)$ の見積法（論文[66]ではガウス過程変換と期待値伝搬法による近似）など、現在も研究が進められている。

8.2.8 RATT 方式

本小節では、Garg 等[76]が提案した RATT 方式による汎化誤差上界の見積法（論文[66]の表 1 には含まれていない）について説明する。RATT 方式では 100%未満の汎化誤差上界の見積りを目標にしており、従来法よりも小さい上界を得ることができる。

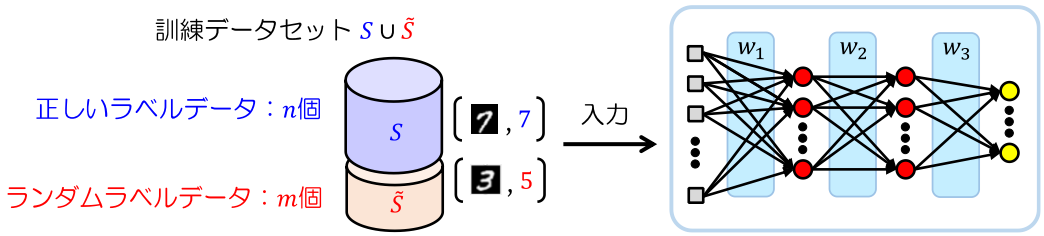


図 8.3 RATT 方式における訓練（ランダムラベルデータセットを用いる）

RATT 方式を適用するためには、図 8.3 に示すように、正しくラベル付けされたデータと共にランダムにラベル付けされたデータも訓練しておく必要がある。ランダムラベルのデータを学習することによって、訓練後の仮説の精度が低下する可能性はあるは、深層学習では、誤りラベルよりも先に正しいラベルのデータに適合する早期学習現象（early learning phenomenon）が実験的にも理論的にも示されており、早期に訓練を打切ることによって、誤りラベルの悪影響を抑えることができる。

まず、次式によって定義される、正しいラベルのデータセットSとランダムラベルのデータセット$\tilde{S}$の両方についての訓練誤差と正則化項の和を最小にする仮説 $\hat{h}$ について考える。

$$\hat{h} \coloneqq \underset{h \in \mathcal{H}}{\operatorname{argmin}} \left(\hat{L}_{S \cup \tilde{S}}(h) + \lambda R(h) \right)$$

ここで、$R(h)$は任意の正則化関数（例えば、hの重みwのL^2ノルムの二乗$\|w\|_2^2$）、λは正則化定数（訓練データセット $S \cup \tilde{S}$ に非依存）である。訓練に用いた正しいラベルデータセットSのサイズをn、ランダムラベルのデータセット$\tilde{S}$のサイズをm、分類クラス数をkとするとき、仮説 $\hat{h}$ の汎化誤差上界は次の不等式によって見積もることができる。この不等式は、確率 $(1-\delta)$ 以上で成り立つことが示されている[76]。

$$L(\hat{h}) \leq \hat{L}_S(\hat{h}) + (k-1)\left(1 - \frac{k}{k-1}\hat{L}_{\tilde{S}}(\hat{f})\right) + \left(\sqrt{k(k-1)} + \sqrt{k} + \frac{m}{n\sqrt{k}}\right)\sqrt{\frac{\ln\frac{4}{\delta}}{2m}}$$

ランダムなラベルのデータ（ランダムなラベル付けでは確率的に正しいラベルが付けられる可能性がある）よりも誤りラベルのデータを用いて訓練した方が、上記の汎化誤差上界の不等式は簡単になる。しかし、誤ったラベルをつける作業には人手が必要であり、その人的コストは低くない。ランダムラベルは自動的に（ランダムに）ラベル付けできるため、実際に RATT 方式を適用するときの作業コストを抑えることができる。

上記の不等式は、次の仮定 1 のもとで深層学習の結果として得られる仮説hに対しても適用可能である。

- [仮定 1] ランダムデータセット$\tilde{S}$に含まれる誤りラベルのデータに対する経験誤差は、$\tilde{S}$に含まれない誤りラベルのデータに対する経験誤差よりも小さい

一般に、訓練データセットに含まれるデータに対する経験誤差は、含まれない経験誤差よりも小さくなるため、訓練の終盤では仮定 1 は成り立つが、早期訓練打切りが早すぎる場合は成り立たない可能性もある。早期学習現象を考慮すると、誤ったラベルのデータに適合する前に訓練を打ち切る必要があるが、仮定 1 が満たされる程度には訓練を継続する必要もある。訓練中にランダムラベルのデータセットに対する経験誤差（ランダムラベルデータへの適合度）を観測し、訓練打ち切り時期を判断するとよい。

Garg 等[76]は RATT 方式を、データセット MNIST（手書き数字の画像）, CIFAR-10（乗り物や動物の画像）, IMDb（映画評価のテキスト）に適用し、100%未満の意味のある汎化誤差上界が得られたことを報告している。訓練時にランダムラベルデータを学習する必要はあるが、ランダムなラベル付けは簡単であり（自動化可能であり）、上記の不等式による汎化誤差上界の見積りも簡単である。ランダムラベルの影響や仮定の妥当性など、より詳しい調査・研究は必要であるが、汎化性能の評価指標のひとつとして、今後検討すべき技術であると考える。

8.3 汎化誤差上界見積法の評価実験

本節では、8.2 節で紹介した汎化誤差上界法のうち二つの手法について評価実験を行った結果を報告する。まず、8.3.1 小節では、最近の汎化誤差見積法の基本となる感度方式（8.2.4 小節にて解説）による実験の結果について報告する。次に、8.3.2 小節では、ランダムラベルを用いた汎化誤差上界見積法の RATT 方式（8.2.8 小節にて解説）による実験の結果について報告する。

8.3.1 感度方式による汎化誤差上界見積り実験

8.2.4 小節で解説した感度方式による汎化誤差上界の見積式（不等式の右辺）は多くの論文で参照されているが、実際にその上界の見積結果について書かれた論文はほとんどない。本節では、感度方式による汎化誤差上界の見積例を報告し、汎化性能の評価指標としての適用可能性について考察する。

感度方式による汎化誤差上界の見積例として、図 8.4 に示す全結合ニューラルネットワークを用いた。このニューラルネットワークは 2 つの中間層をもつ 4 層の順伝播型であり、各層のニューロン数は、入力層 28 × 28個（784 個）、中間層 500個、出力層 10 個である。中間層のニューロンの活性化関数に ReLU、出力層では Softmax を使用している。

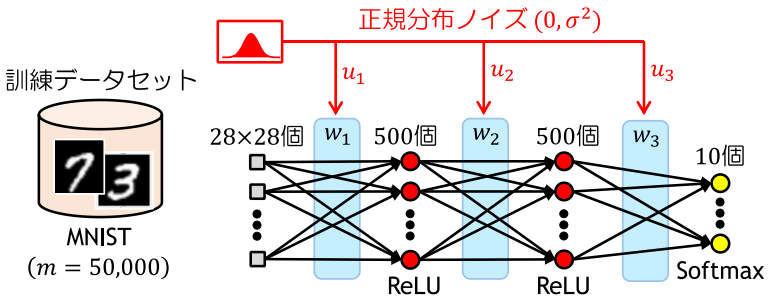
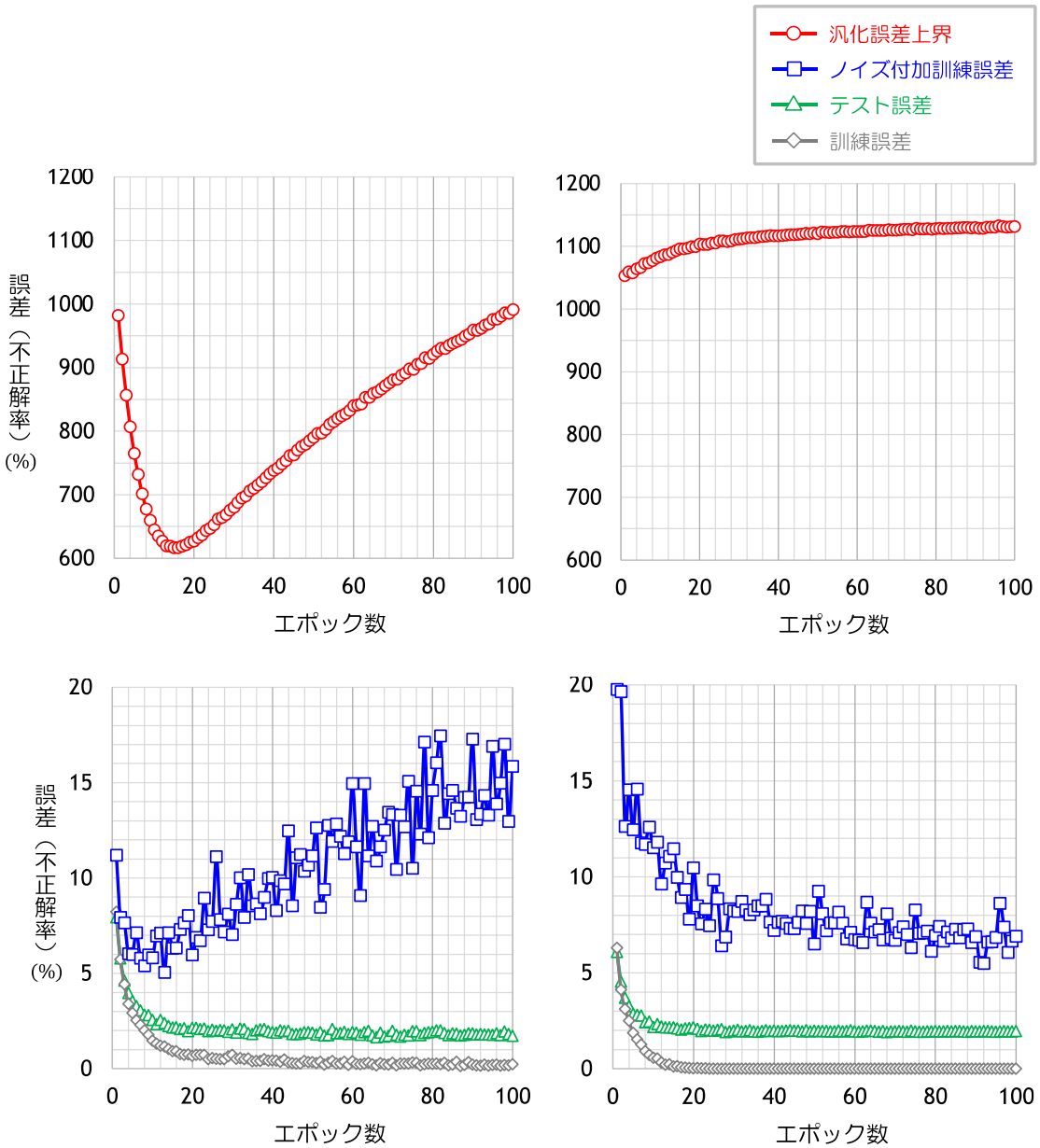


図 8.4 感度方式の実験用ニューラルネットワークの構成

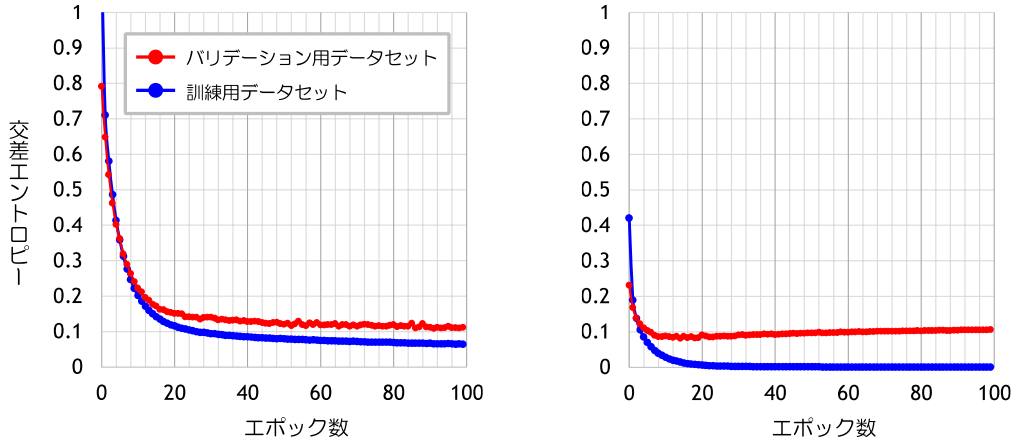
実験には、手書き数字の画像のデータセット MNIST（28 × 28画素、輝度[0,1]）を使用した。データセットのサイズは訓練用50,000、バリデーション用5,000、テスト用10,000である。訓練に用いたアルゴリズムは確率的勾配降下法であり、正則化有り（L^2ノルムの正則化係数 0.001）と正則化無しの場合の 2 通りの実験を行った。

図 8.5 に訓練中の感度方式による汎化誤差上界（$\delta = 0.1$、信頼度90%）の見積り結果を示す。また、補足情報として、図 8.5 の訓練中の交差エントロピーを図 8.6、重みのL^2ノルムを図 8.7 に示す。使用したノイズの標準偏差σは、予備実験によって得られた、ノイズ付加訓練誤差が約10%をなったときの値であり、図 8.5 (a)の正則化有り（L^2係数 $= 0.001$）の場合は $\sigma = 0.03$、図 8.5 (b)の正則化無し（L^2係数 $= 0$）の場合は $\sigma = 0.05$である。



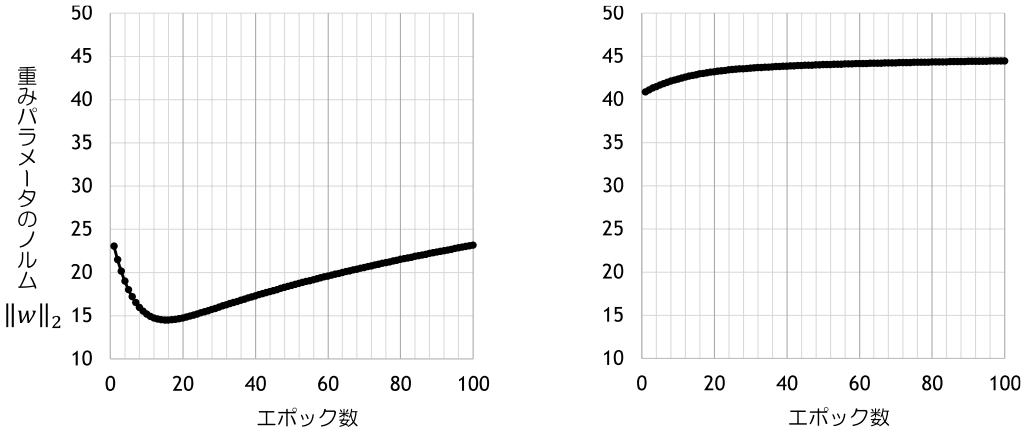
(a) 正則化有り（L_2係数 $= 0.001, \sigma = 0.03$）　(b) 正則化無し（L_2係数 $= 0, \sigma = 0.05$）

図 8.5　感度方式による汎化誤差上界の見積結果（訓練中）



(a) 正則化有り（L_2係数 $= 0.001, \sigma = 0.03$） (b) 正則化無し（L_2係数 $= 0, \sigma = 0.05$）

図 8.6 訓練過程における交差エントロピーの変化



(a) 正則化有り（L_2係数 $= 0.001, \sigma = 0.03$） (b) 正則化無し（L_2係数 $= 0, \sigma = 0.05$）

図 8.7 訓練過程における重みノルム$\|w\|_2$の変化

図 8.5 には、正則化有りの場合と正則化無しの場合の見積結果を示しているが、まずは正則化有りの場合(a)について考察する。図 8.5 (a)に示すように、汎化誤差上界は982%（1 エポック）から617%（15 エポック）まで低下し、その後は再び増加している。16 エポック以降の汎化誤差上界の増加は、訓練データセットへの過剰適合（overfitting）が原因であると考えられるが、図 8.5 (a)と図 8.6 (a)に示すように、16 エポック以降も、テスト誤差と交差エントロピーは減少傾向が続いている。一方、図 8.7 (a)に示すように、重みのノルム$\|w\|_2$は 15 エポックのときに最小値（14.5）となり、16 エポック以降増加している。不等式の右辺第 2 項は$\|w\|_2$とともに増加するため、汎化誤差上界が 16 エポック以降増加する主な原因は$\|w\|_2$の増加であることがわかる。ただし、図 8.5 (a)のノイズ付加訓練誤差が 16 エポック以降増加傾向にあることには注目すべきことである（ノイズの大きさは固定：標準偏差 $\sigma = 0.03$）。一般に、重み$\|w\|_2$のノルムの増加は（重みに付加される）ノイズの影響を小さくする傾向にある。16 エポック以降、重みパラメータのノルム$\|w\|_2$（図 8.7 (a)）は増加しているにもかかわらず、ノイズ付加訓練誤差（図 8.5 (a)）も増加していることは、ノイズに対する耐性が低下していることを意味する。テスト誤差や交差エントロピーでは観測さ

れていないが、これも訓練データセットへの過剰適合の一種ではないかと考えられる。

次に、正則化無しの場合（図 8.5 (b)）について説明する。図 8.5 (b)に示すように、汎化誤差上界は1,053%（1 エポック）から1,131%まで（100 エポック）まで徐々に増加している。訓練誤差は 33 エポック以降ほぼ 0 であり、それ以降に大きな変化はない。正則化ありの場合（図 8.5 (a)）と比較して、図 8.5 (b)のノイズ付加訓練誤差（ノイズの大きさは固定：標準偏差 $\sigma = 0.05$）は小さく、ノイズに対して耐性があるようにみえるが、これは図 8.7 (b)に示すように、重みパラメータの絶対値が大きいためであり、正則化無しの場合の方が汎化誤差上界は大きくなる。この結果は、正則化が汎化性能の向上（汎化誤差の低下）に効果があることと一致している。

感度方式による汎化誤差上界の見積結果は100%を超えることが多く、本小節の実験でも100%未満になることはなかった。そのため、感度方式による汎化誤差上界を汎化性能の絶対的な評価指標として適用することは難しい。一方で、既存研究[68][74]でも報告されているように、感度方式の汎化誤差上界は、汎化性能の相対的な評価には有効であるとの結果が得られた。特に、図 8.5(a)の汎化誤差上界の変化（16 エポック以降の増加）は、データサンプルによる従来の評価（テスト誤差やバリデーションデータセットによる交差エントロピー）では観測できないノイズに対する耐性の低下（汎化性能の低下）を示した。今回の簡易な実験で結論づけることはできないが、感度方式による汎化誤差上界は、データサンプルによる評価とは異なる視点での汎化性能の評価指標のひとつになりうると考えられる。

8.3.2 RATT 方式による汎化誤差上界見積り実験

図 8.8 に示すように、RATT 方式による汎化誤差上界の実験でも、8.3.1 小節の感度方式の実験と同じ全結合ニューラルネットワーク、データセット MNIST、確率的勾配降下法（正則化：有/無）を用いた。データセットのサイズは、訓練用 48,000（正しいラベルデータ 40,000、ランダムラベルデータ 8,000）、バリデーション用 4,000、テスト 10,000 である。重みの初期化には正規分布$\mathcal{N}(0, 0.1^2)$の乱数を用いた。

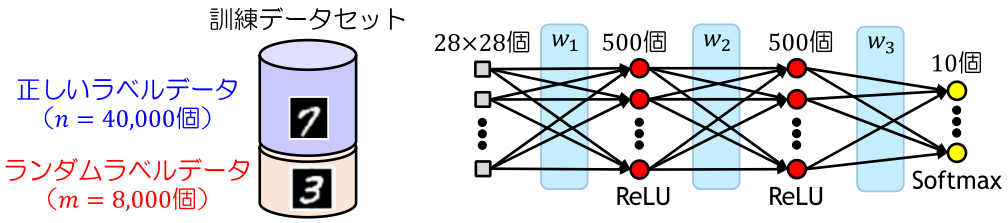
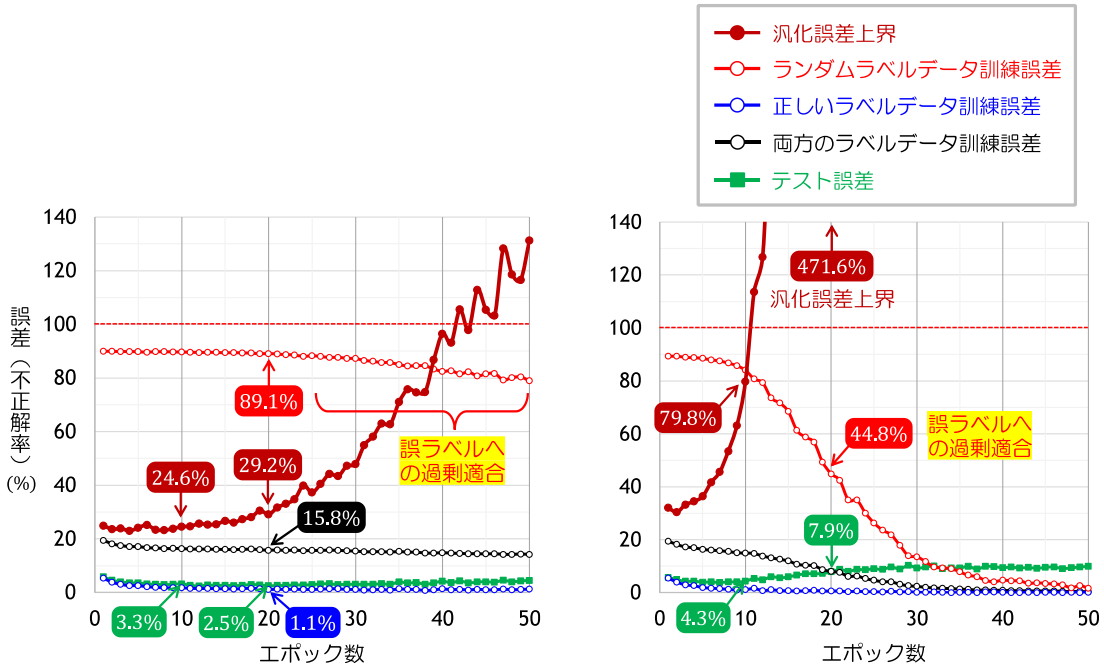


図 8.8 RATT 方式の実験用ニューラルネットワークの構成

図 8.8 のニューラルネットワーク訓練中の、RATT 方式による汎化誤差上界の見積結果（$\delta = 0.1$、信頼度90%）を図 8.9 に示す。まず、正則化有りの場合の見積結果について考察する。図 8.9 (a)にみられるように、RATT 方式では、100%未満の汎化誤差上界が得られている。例えば、10 エポックのときの汎化誤差上界は24.6%、20 エポックのときは29.2%の汎化誤差上界である。そのときのテスト誤差が、各々、3.3%と2.5%であることを考慮すると、まだ高めの汎化誤差上界であるが、他の汎化誤差上界見積法と比較すると、意味のある低めの上界が得られている。



(a) 正則化あり（L^2係数 = 0.001）　(b) 正則化なし（L^2係数 = 0）

図 8.9　RATT 方式による汎化誤差上界の見積結果（訓練中）

図 8.9 (a)のランダムラベルデータセットについての訓練誤差に着目すると、20 エポック程度までは90%を維持し、それ以降、徐々に低下している。これは早期学習現象によるものと考えられる。MNIST のクラス数は 10 個であり、ランダムにラベル付けした場合も10%程度は正しいラベルが付けられる可能性がある。早期学習現象は、誤りラベルのデータよりも先に正しいラベルのデータに適合する現象であり、20 エポック程度までは正しいラベルデータのみに適合が行われたため、ランダムラベルデータの訓練誤差は90%程度になっていると考えられる。20 エポック以降は、徐々に誤ったラベルデータ（ランダムラベルのデータの約90%）にも適合していくため、ランダムラベルデータの訓練誤差は徐々に低下し、テスト誤差は徐々に増加している。

次に、正則化無しの場合の見積結果について考察する。図 8.9 (b)のランダムラベルデータセットについての訓練誤差は 10 エポック程度までは90%程度を維持しているが、それ以降は急速に減少している。これは、正則化なしの場合も早期学習現象はみられるが、誤ったラベルのデータに早い段階から適合することを意味している。このため、正則化なしの場合は汎化性能の低下が想定され、実際、10 エポック以降の汎化誤差上界は100%を超えている。

正則化の有無による汎化性能を比較するため、次のように図 8.9 (a)と(b)の 10 エポックの汎化誤差上界に着目する。

- 正則化有りの場合：テスト誤差 3.3%, 汎化誤差上界 24.6%
- 正則化無しの場合：テスト誤差 4.3%, 汎化誤差上界 79.8%

正則化が無い場合は有る場合と比較して、10 エポックのときの二つの仮説（学習モデルの入出力関数）のテスト誤差の差は1%であるが、汎化誤差上界については 3 倍以上異なる。このことは、テスト誤差では評価が難しい汎化性能の違いを、汎化誤差上界によってより明

確に評価できる可能性を示している。

本小節の MNIST を用いた実験では RATT 方式によって100%未満の汎化誤差上界を見積もることができた。訓練にランダムデータを含めるため、訓練後の仮説の精度（正解率）低下が懸念されるが、早期学習現象によって、誤りラベルデータへの適合は正しいデータへの適合よりも遅くなる傾向がある。すなわち、早期訓練打切りによって、適度な精度と意味のある（100%未満の）汎化誤差上界を両立する仮説を得ることも可能であると考えられる。ただし、8.2.8 小節で説明した「仮説 1 」を満たすためには、ランダムラベルデータにある程度適合する（学習する）必要があるため、早すぎる訓練打切りは仮説 1 を満たさない可能性がある。例えば、図 8.9 (a)では、ランダムラベルデータセットの訓練誤差が90%より少し低くなっている 20 エポック程度での訓練打切りが妥当であると考えられる。

図 8.9 の実験結果では、正則化による汎化性能への影響を定量的に評価することができた。今後、さらなる調査・実験が必要ではあるが、100%未満の汎化誤差上界も得られており、絶対値としての汎化性能の評価指標のひとつになりうると考えられる。

8.4 まとめ

本調査の結果として、既存手法による汎化誤差上界の見積り結果は 100% を超える無意味な値になることが多く、汎化性能の評価指標として適用することは、現時点ではまだ難しいことがみえてきた。その理由として、汎化誤差の研究の多くが、「訓練パラメータ数が訓練データ数より非常に多くても深層学習では汎化性能が得られる現象」の理論的な解明等を目標にしており、汎化性能の評価を目標にはしていなかったことが挙げられる。汎化誤差上界の見積り結果が 100%を超える場合でも、相対的な汎化尺度として複数の仮説の汎化性能の比較評価には有効であり、評価実験も行われている[68][74]。実際、8.3.1 小節の感度方式の実験では、図 8.5 にみられるように上界の見積結果は100%を超えているが、正則化無しと比較して正則化有りの方が明確に低い汎化誤差上界を示している。また、図 8.5 (a)では、20 エポック以降にノイズに対する耐性の低下が観測されている。テスト誤差は 100 エポックまで減少傾向が続いているが、おそらくこの原因は過剰適合（overfitting）であり、データサンプルからは観測が難しい汎化性能の低下を示していると考えられる。

一方、2017 年頃から汎化誤差上界を 100% 未満に抑えることを目標にした論文（[66][70][75][76]等）も発表されるようになってきており、汎化誤差上界の見積り精度は改善されつつある。実際、8.3.2 小節の RATT 方式[76]の実験では、図 8.6 にみられるように 100%未満の汎化誤差上界が得られており、汎化性能の絶対的な評価指標として利用できる可能性もみえてきた。また、PAC-Bayesian を基礎にして、汎化誤差上界を下げるように重みの分布（平均と標準偏差）を最適化する方法も提案されている[77][78]（詳細は現在調査中）。今後、汎化誤差上界も機械学習モデルの汎化性能の評価指標のひとつになりうると考えている。

9 敵対的データ検出技術

9.1 研究概要

与えられた入力画像が敵対的データ（Adversarial Example）であるかを判別する方法を実用的に確立することを目標として、敵対的データを生成する攻撃と検出手法について、下記の点に着目して代表的な技術の調査および実装を実施している。

- 敵対的データ検出プログラムコードの裏付けと計算実験による確認
- 敵対的データ検出手法の論文の実験結果の再現
- 機械学習システム品質評価テストベッドへの敵対的データ検出フレームワークの構築

敵対的データ検出（Adversarial Examples Detection）とは、与えられた入力の中から敵対的データを検出することであり、既存の最先端の敵対的データ検出方法は次の 4 つの主要なカテゴリに分類できる。

① メトリックベースアプローチ（例. [79]）
② ディノイザーアプローチ（例. [80]）
③ 予測不整合ベースアプローチ（例. [81]）
④ ニューラルネットワーク不変式チェック（NIC）アプローチ（例. [82]）

本章では、これら①～④の各アプローチに基づく敵対的データ検出手法を比較・評価するために追試実験を行った結果について報告する。論文[82]に報告されているように、④のアプローチ（NIC：Neural Network Invariant Checking）が①～④の中で最も高い検出率を示すことを確認できた。この追実験において、①～③については公開されている実装コードを使用したが、④については実装コードが公開されていなかったため、論文[82]にしたがって NIC を実装して追実験を行うとともに、NIC に基づく敵対的データ検出を行うための NIC フレームワークを構築した。そのため、本章では主に④の NIC について説明する。

以降、4 つのアプローチの概要を説明したのち、NIC による敵対的データの検出方法を説明して、その実装方法について述べる。次に、各アプローチの追試実験と NIC による実験の結果を示し、最後に NIC フレームワークの実装とその有効性評価結果について報告する。

9.2 敵対的データ検出アプローチの概要

以下、最先端の敵対的データ検出のための 4 つのアプローチの概要について説明する。

9.2.1 メトリックベースアプローチ

入力（および各ニューロンの出力）の統計的測定を実行して、敵対的データを検出する方法であり、Ma 等は、最近、Local Intrinsic Dimensionality（LID）と呼ばれる測定の使用を提案した[79]。この方法では、サンプルの距離分布と個々のレイヤーの近隣の数を計算することによって、サンプルを囲む領域の空間充填能力を評価する LID 値を推定し、敵対的データが大きな LID 値を持つ傾向がある性質を用いて、敵対的データを検出している。LID は、敵

対的データの検出に対して、従来のカーネル密度推定（KD）やベイジアン不確実性（BU）よりも優れており、現在この種の検出器の最先端の技術となっている。

9.2.2 ディノイザーアプローチ（**Denoiser**、ノイズ除去）

各入力に対して前処理ステップでノイズを除去することによって敵対的データを検出する方法である。この方法では、学習モデル内の主要なコンポーネントを強調できるように、学習モデルまたはノイズ除去器（エンコーダーおよびデコーダー）をトレーニングして画像をフィルター処理する。このフィルターを用いて、攻撃者が敵対的データを生成するために追加したノイズを除去し、誤分類を修正することができる。MagNet[80]は、検出器とリフォーマー（トレーニング済みの自動エンコーダーと自動デコーダー）を使用して、敵対的データを検出する方法である。

9.2.3 予測不整合ベースのアプローチ（**Prediction inconsistency based approach**）

元のニューラルネットワークと人間の知覚可能な属性で強化されたニューラルネットワークとの間の不一致を測定して、敵対的データを検出する方法である。この方法の最先端の検出技術であるフィーチャスクイージング（Feature Squeezing）[81]は、さまざまな攻撃に対して非常に高い検出率を実現することができる。フィーチャースクイージングは、ディープニューラルネットワーク DNN の不必要に大きな入力特徴空間によって攻撃者が敵対的データを生成できることに着目しており、勾配ベースの攻撃の検出に焦点を当てている。フィーチャースクイージングによる敵対的データの検出手順を以下に示す。

1. 元の入力画像にスクイージング技術（画像の色深度を減らし、画像を平滑化する技術）を適用して複数のスクイーズ画像を生成する。
2. 元の入力画像と複数のスクイーズ画像をディープニューラルネットワークに入力し、入力画像の推論結果（予測ベクトル）と各スクイーズ画像の推論結果との距離を測定する。
3. 元の入力画像とスクイーズ画像の差（距離）の一つがしきい値を超える場合に、元の入力画像を敵対的データとして検出する。

9.2.4 ニューラルネットワーク不変性チェック（**NIC**）アプローチ

NIC[82]では、ニューラルネットワーク内部の値の不変量（VI: Value Invariants）と来歴不変量（PI: Provenance Invariants）に着目する。値の不変量 VI は各層の可能なニューロン値の分布であり、来歴不変量 PI は 2 つの連続した層の可能なニューロン値パターン（2 層にわたるフィーチャ間の相関の要約）である。ある入力がこれらの不変量に違反している場合に、その入力は敵対的データとして検出される。それらの不変量 VI と PI を良性の入力データで訓練し、敵対的データを検出する 1 クラス分類（OCC）問題としてモデル化する。上で説明した①～③に基づく手法よりも高い検出率が報告されている[82]。以降、NIC のシステム設計概要と実装について、各々9.3 節と 9.4 節で詳しく説明する。

9.3 NIC のシステム設計概要

NIC の検出器の構築と検出の手順（ステップ A～C：訓練時、D～E：実行時）を図 9.1 を用いて説明する[82]。この不変量 VI, PI の訓練では、敵対的ではない良性のデータのみを使用する。

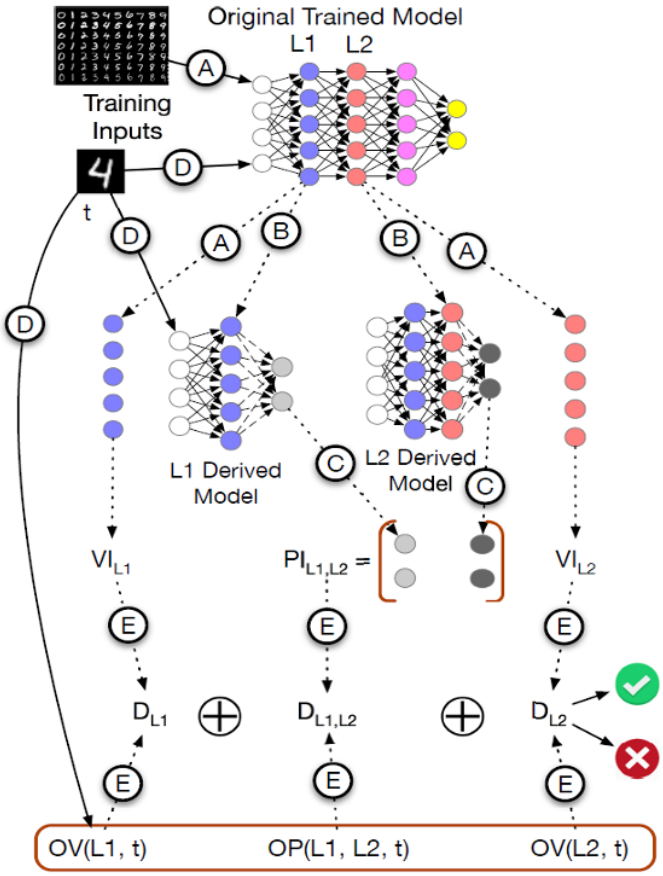


図 9.1 システム設計概要（論文[82]の図 8）

- **ステップ A**：各訓練データ入力の各層で各ニューロンの出力値を収集する。
- **ステップ B**：各層 k（例えば、L1、L2）に対して、入力層から k 層までのサブモデルを抽出し、元のモデルと同じ出力ラベルをもつ新しい softmax 層を追加して、派生モデル（図 9.1 中の DerivedModel）を作成する
- **ステップ C**：すべての派生モデルに対して各良性訓練データを入力し、これらのモデルの最終出力（すなわち、個々のクラスの出力確率値）を収集する。連続する層の組ごとに、この派生モデルの分類結果の分布を用いて訓練する。この訓練された分布が、これらの 2 つの層の PI となる。
- **ステップ D**：各テストデータ t（例えば、図 9.1 中の“4”の画像）を、元のモデルの他に、すべての派生モデルにも入力して、元のモデルの各層の活性化値を観測値 OV（例えば、図 9.1 中の OV(L1,t)）と、連続した層の派生モデルの分類結果（組）を収集す

る。この分類結果から観測された出処 OP（例えば、OP(L1,L2,t)など）を得る。

- **ステップ E**：OV と OP が対応する VI と PI の分布に適合する確率 D を計算する。入力 t が敵対的である可能性を、これらの D 値をすべて集約して同時予測する。

9.4 NIC のシステム実装

NIC に基づいて敵対的データを検出するために、PI と VI から直和空間（ベクトル）を構成し、このベクトルを分類するための OSVM（One Class Support Vector Machine）を構築する。訓練済みの DNN（Deep Neural Network）のモデル（以降、これを M と記述する）の層 l に対する入力を x_l とするとき、層 l の出力 f_l は次式により与えられる。

$$f_l = \sigma(x_l \cdot w_l^T + b_l)$$

ここで、σ は層 l の活性化関数、w_l^T は重み行列、b_l はバイアスである。このとき、VI と PI、および OSVM で分類する直和空間は次のように求められる。

- **VI の計算**：モデル M の各層 l の VI は以下の最適化問題を解いて決定する。

$$VI_l = \min\left[\sum_{x \in X_b} J(f_l \circ f_{l-1} \circ \cdots \circ f_1(x) \cdots w^T - 1)\right]$$

ここで、J はエラー評価関数、X_b は M を作成する際に使用したバッチである。また、$\circ$はモノイドであり、この場合では、f_k をベクトル化したものである。

- **PI の計算**：$PI_{l,l+1}(x)$は層 l および層$l+1$ の派生モデルの分類出力に基づいて、x が良性である（敵対的でない）確率は、次の最適化問題を解いて推測する。

$$PI_{l,l+1}(x) = \min\left[\sum_{x \in X_b} J(concat(D_l(x), D_{l+1}(x)) \cdots w^T - 1)\right]$$

ここで、層 l の派生モデルD_l は、層 l の後に softmax 層を追加してのように定義される。

$$D_l = \text{softmax} \circ\ f_l \circ f_{l-1} \circ \cdots \circ f_1$$

- **PI と VI の直和空間**：上記の最適化により求めた VI と PI から、モデル M の学習データのバッチごとに以下の直和空間（ベクトル）を作成する。

$$VI_1 \oplus PI_{1,2} \oplus VI_2 \oplus PI_{2,3} \cdots VI_B \oplus PI_{B-1,B} \oplus VI_B$$

このベクトルは $L \times 3$ 次元（L は M の層数）であり、これは個数B のベクトル空間（直和空間）になる。NIC ではこの空間に対して OSVM を行う。

9.5 計算機実験

敵対的データ検出技術（NIC)の効果を確認するため、下記の実験環境で、論文[82]の実験の追試を行なった。

- ハードウェア環境：産総研 ABCI[83]
- データセット：画像分類の実験には、MNIST[84]、CIFAR-10[85]の 2 つの一般的な画像データセットを用いた。MNIST は手書き数字認識に使用されるグレースケール画像データセットであり、CIFAR-10 はオブジェクト認識に使用されるカラー画像データセットである。なお、NIC に対しては、LFW（顔画像）[86]についても実験を行った。
- 攻撃：敵対的データの生成には、非標的型攻撃（FGSM L^2, L^∞）、標的型攻撃 JSMA、勾配ベースの攻撃（CW L^2）の方法を使用した。FGSM と JSMA の実装には、Cleverhans ライブラリ[87]を使用した。

最初に、①～③の各アプローチに基づく敵対的データ検出手法を評価するため、LID[79]、MagNet[80]、フィーチャスクイージング[81]の公開されている実装コードを用いて、各論文の追試実験を行った。その結果、各論文に報告されている検出率を確認でき、この 3 つの中では、フィーチャスクイージングが最も高い検出率を示していた。

次に、④のアプローチに基づく敵対的データ検出手法を評価するため、9.4 節で実装した NIC のコードを用いて実験を行った。表 9.1～表 9.3 に、各々、MNIST、CIFAR-10、LFW のデータセットに対する敵対的データ検出計算実験の結果示す。ここで、正答率は、9.4 節で説明した分類器（OSVM）に敵対的データを入力し、敵対的データであると判定された割合である。なお、実験に使用した CNN モデルは LeNet5、OSVM の Kernel は RBF（MNIST：$\gamma = 0.1\sim0.27$, CIFAR-10：$\gamma = 0.11\sim0.2$, LFW：$\gamma = 0.005\sim0.90$）である。本実験結果では、論文[82]で報告されていたデータセットや攻撃方法だけでなく、報告されていないデータセット LFW、攻撃方法（FGSM L^∞）についても高い検出性能を確認することができた。

表 9.1　MNIST データセットに対する敵対的データ検出計算実験結果

Data Set	Attack	Invariant	正答率	データ件数	論文[82]正答率
MNIST	FGSM L^2	VI	97%	2800	100%
		PI	98%		84%
		NIC	97%		100%
	FGSM L^∞	VI	98%	2800	—
		PI	98%		—
		NIC	98%		—
	JSMA	VI	100%	280	83%
		PI	100%		100%
		NIC	100%		100%
	CW2	VI	100%	280	95%
		PI	100%		96%
		NIC	100%		100%
	Trojan	VI	100%	3200	100%
		PI	100%		100%
		NIC	100%		100%

表 9.2 CIFAR-10 データセットに対する敵対的データ検出計算実験結果

Data Set	Attack	Invariant	正答率	データ件数	論文[82]正答率
CIFAR-10	FGSM L^2	VI	99%	6400	100%
		PI	99%		52%
		NIC	99%		100%
	FGSM L^∞	VI	100%	6400	—
		PI	100%		—
		NIC	100%		—
	JSMA	VI	97%	320	62%
		PI	95%		100%
		NIC	96%		100%
	CW2	VI	98%	320	88%
		PI	95%		89%
		NIC	96%		100%
	Trojan	VI	100%	3200	100%
		PI	100%		100%
		NIC	100%		100%

表 9.3 LFW データセットに対する敵対的データ検出計算実験結果

Data Set	Attack	Invariant	正答率	データ件数	論文[82]正答率
LFW	FGSM L^2	VI	98%	28222	—
		PI	98%		—
		NIC	98%		—
	FGSM L^∞	VI	100%	2822	—
		PI	100%		—
		NIC	100%		—
	JSMA	VI	100%	280	—
		PI	100%		—
		NIC	100%		—
	CW2	VI	100%	840	—
		PI	100%		—
		NIC	100%		—
	Trojan	VI	100%	3200	—
		PI	100%		—
		NIC	100%		—

9.6 NIC フレームワークの実装

9.5 節で NIC の有効性の確認を目的とした計算機実験を行うため、簡易に、9.3 節と 9.4 節に基づき NIC 法の実装をおこなった、その際、原論文[45]に実装上の問題点などが明らかとなった。これら問題点を明らかにしつつ、敵対的データ（Adversarial Examples）に対する脆弱性をベンチマークする環境（攻撃・防御・検出）を構築し、敵対的データの高い検出率を目標とする NIC フレームワークをテストベッドに構築するため、アルゴリズムの再検討を行なった。

9.6.1 NIC フレームワークの概要

NIC フレームワークは、各層からの出力取出し、正常データの VI, PI 計算、敵対的データの VI, PI, NIC の計算、OSVM の評価と結果の表示の 5 つのパートで構成されている。NIC フレームワークのユースケースを図 9.2 に示す。また、敵対的データを検出する処理手順を図 9.3 に示す。

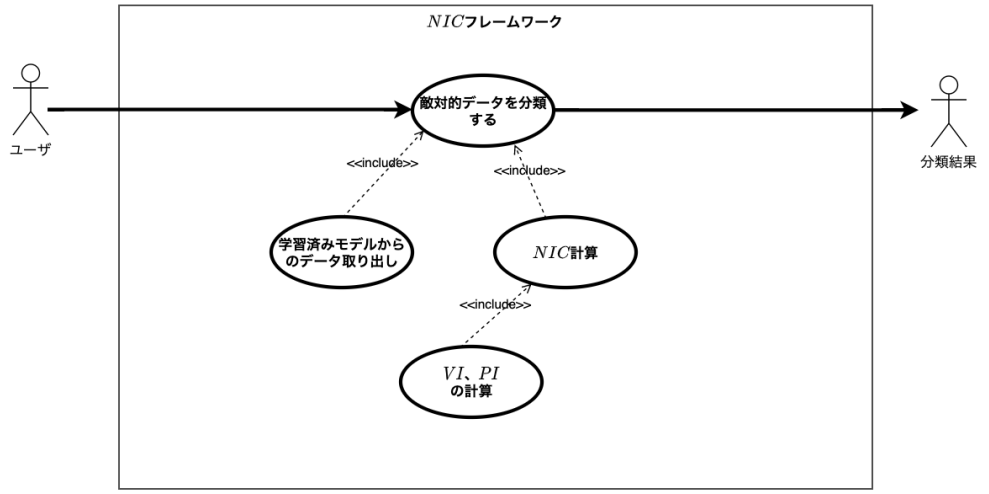


図 9.2 NIC フレームワークのユースケース

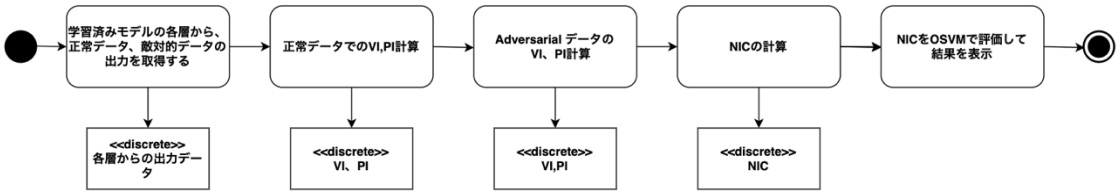


図 9.3 NIC フレームワークによる敵対的データ検出の処理手続き

図 9.3 に示すように、NIC フレームワークによる敵対的データ検出全体の処理手続きは 5 つのパートから構成されている。各パートの機能（入力、処理、出力）を表 9.4 に示す。

表 9.4 敵対的データ検出処理手続きを構成する各パートの機能

各層からの出力取り出し	
入力	正常データ（画像）
	敵対データ（画像）
	正常データで学習した、学習済みモデル（正常データで学習したモデル）
処理	正常データ、敵対的データの学習済みモデルの各層の出力を取得して、「numpy の npy」形式で保存する。
出力	各層からの出力データ

正常データの VI, PI 計算	
入力	正常データの各層からの出力データ
処理	正常データの各層の出力データから VI、PI を計算する。
出力	VI, PI

敵対的データの VI, PI 計算	
入力	敵対的データの各層からの出力データ
	正常データで PI に計算時に作成された PI の派生モデル
処理	敵対的データの各層の出力から VI, PI を計算する。
出力	VI, PI

NIC の計算	
入力	正常データの VI, PI
	敵対的データの VI, PI
処理	正常データの VI, PI から正常データの NIC を作成、敵対的データの VI, PI から敵対的データの NIC をそれぞれ計算する。
出力	正常データの NIC、敵対的データの NIC

OSVM での評価と結果の表示	
入力	正常データの NIC
	敵対的データの NIC
処理	正常データで OSVM を学習させてモデルを作成、この学習済みモデルを使用して敵対的データを判定する。OSVM は sk-learn の one class svm API を使用している。その後、判定結果を表示する。
出力	評価結果

9.6.2 **OSVM** 評価結果の出力

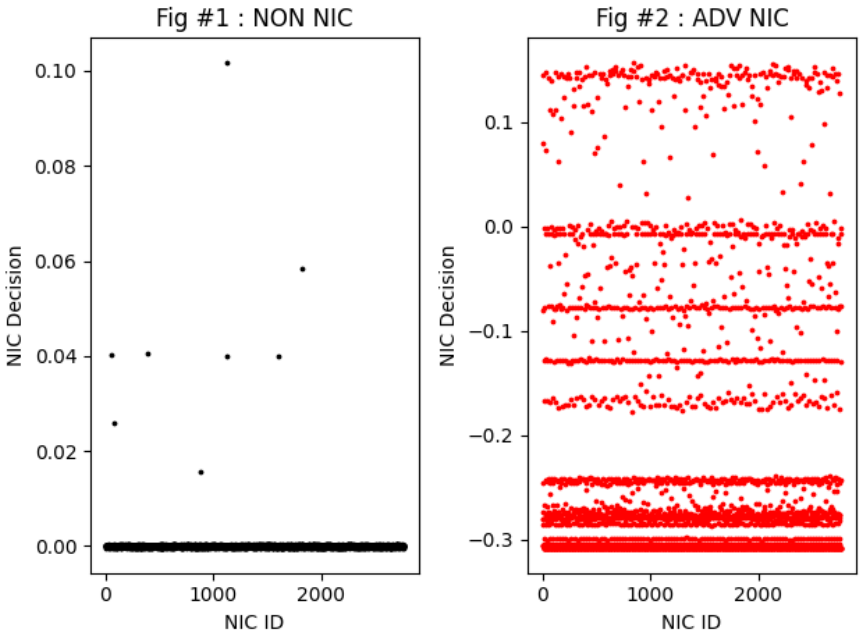
NIC フレームワークの実装には Python の機械学習のライブラリである scikit-learn を用いた。例えば、図 9.3 の最終パートである OSVM は scikit-learn の OneClassSVM の class を用いて次のように実装した。

```
class sklearn.svm.OneClassSVM(array, kernel='rbf', gamma='auto', nu=0.3)
```

ここで、各引数の意味は次の通りである。

- array：正常データの NIC でパラメータを学習し、敵対的データの NIC で推定する
- kernel：One Class SVM のアルゴリズムとして、RBF カーネルを使用する
- gamma：RBF カーネルの γ パラメータ、「auto」を指定
- nu：学習誤差の割合の上限とサポートベクトルの割合の下限を指定する（今回は 0.3）

NIC フレームワークに 1 個の正常データと、その敵対的データを入力したときの、各層からの出力値を図 9.4 に示す。ここで、横軸は各層で NIC を計算するために導出したモデルの ID（注. 例えば CNN の Convolution 層など、1 画像に対する各層からの出力は複数存在する）、縦軸は正常データの NIC の One Class SVM 分類超平面までの各 NIC の符号付き距離、すなわち、この場合、正常データへの近さを表している。図 9.4(a)の黒点が正常データに対する出力、図 9.4(b)の赤点が敵対的データに対する出力である。なお、図 9.4(b)の敵対的データの生成には FGSM L^{∞}の攻撃方法を用いた。



(a) 正常データ入力　　(b) 敵対的データ入力

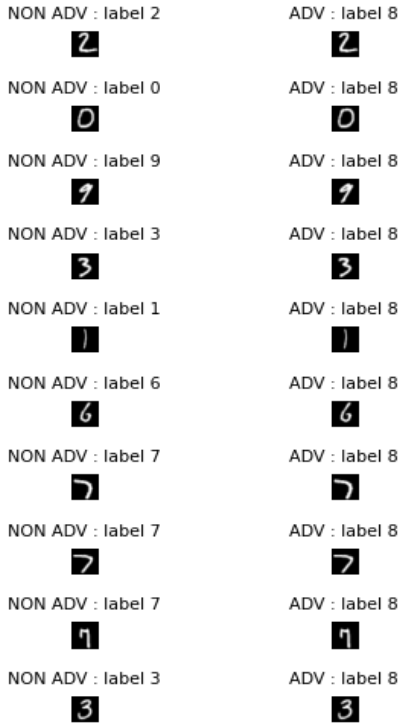
図 9.4　NIC フレームワークの出力比較

正常データで One Class SVM を学習させた場合、One Class SVM の関数 $f(x)$ によって、$f(x) \geq 0$ ならば正常、$f(x) < 0$ ならば異常であると判断することができる。図 9.4 の結果は正常データを入力したときの出力がほぼ 0 の周辺に集中している（図 9.4(a)）のに対して、敵対的データを入力したときは約 94%が 0 未満となっている（図 9.4(b)）。この結果から NIC による敵対的データ検出の有効性を確認できる。

9.6.3　敵対的データの作成

図 9.3 に示すように、NIC フレームワークには敵対的データの作成プログラムは含まれな

い。敵対的データを作成する場合は CleverHans [88]の使用を推奨する。図 9.5 に、手書き数字画像（MNIST）の正常データとその画像から生成した敵対的データ（攻撃方法：FGSM L^2）および各画像データに対して推論されたラベルを示す。図 9.5(b)の推論結果（label 8）にみられるように、生成された敵対的データは全て 8 と誤判断されている。



(a) MNIST の元データ　　(b) 生成した敵対的データ

図 9.5　MNIST（手書き数字）画像からの敵対的データ生成例とその判定結果

9.6.4　VI、PI、NIC の計算コスト削減

原論文[75]の VI, PI, NIC の計算方法は 9.4 節で説明したが、そのまま計算するとそれぞれのデータ（ベクトル）の次元が非常に大きくなり、いわゆる「次元の悪魔」に囚われることになるため、なるべく次元が低下するように工夫している。以下、各計算の簡素化の方法について説明する。

- **VI の計算**：NIC フレームワークでは、入力データ（正常、敵対的双方）と、VI, PI, NIC との対応を明確にするため（検証の正確性を記すため）$X_B = 1$とする。また入力データは全て正規化して計算しているため、下記のように簡素化した式を使用している。

$$VI_l = f_l \circ f_{l-1} \circ \cdots \circ f_2 \circ f_1$$

- **PI の計算**：上記の VI と同様に、$X_B = 1$として計算している。従って下記のように簡素化した式を使用している。

$$PL_{l,l-1} = concat(D_l, D_{l-1}) \circ \cdots \circ concat(D_2, D_1)$$

・ **NIC の計算**：次元抑制のため $X_B = \left(\text{出力を取得した層の数}\right)$ としている。

9.7 Kullback-Leibler 情報量による NIC の有効性評価

Kullback-Leibler 情報量を用いて、正常データと敵対的データの画像、及び、NIC の乖離度を計算し、NIC の有効性を評価した結果について報告する。

9.7.1 Kullback-Leibler 情報量

Kullback-Leibler 情報量は 2 つの確率分布P（確率密度関数p）、Q（確率密度関数q）の間の乖離度を測る尺度である。ただし、厳密な距離の公理を満足しないので、厳密な意味での距離にはならない。Kullback-Leibler 情報量（以後、$KL(P \parallel Q)$と表す）は次式で定義される。

$$KL(P \parallel Q) = \int p(x) \log \frac{p(x)}{q(x)}$$

両者が同じ場合には 0 、乖離が進むと値が大きくなる（log があるため収束は保証されない）。図 9.6 に Kullback-Leibler 情報量の簡単な計算例を示す。図 9.6 左は、PとQ双方とも平均 0.5、分散 0.5 の正規分布の $KL(P \parallel Q)$ で、値は 0 である。右は、Pが平均 0.5、分散 0.5、Q が平均 0.55、分散 0 .55 の $KL(P \parallel Q)$ で、値は 0.053 である。

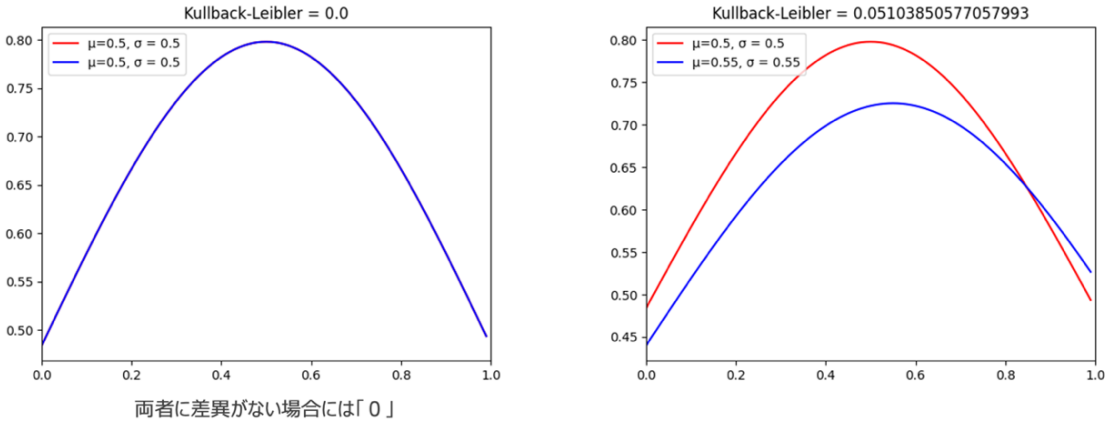


図 9.6 Kullback-Leibler 情報量の計算例

9.7.2 Kullback-Leibler 情報量の推定

Kullback-Leibler 情報量は比較する確率分布が確定していることが前提であるが、実際には、正常データ、敵対的データ双方ともに単なる画像の集合であり、分布は明確でない。しかし、確率分布が明確でない集合同士の Kullback-Leibler 情報量を近似する手法[89]が知られている。その近似計算の概略は、密度比 $r(x) = p(x)/q(x)$ を Kullback-Leibler 情報量を最小化することを制約として、θの線形多項式 $r_\theta(x)$ を最適化問題として解くことにある。

$$r_\theta(x) = \sum_{j=1}^{b} \theta_j \psi_j(x) = \boldsymbol{\theta}^T \boldsymbol{\psi}(x)$$

ここで、次式により定義される$\psi_j(x)$は RBF カーネルである。

$$\psi_j(x) = \exp\left(-\frac{\|x - x'\|^2}{2h^2}\right)$$

ここで、hは決定可能な定数でバンド幅である。

このとき、上記の線形多項式 $r_\theta(x)$ を用いて、Kullback-Leibler 情報量を求めるための目的関数は次式により与えられる。

$$\min_\theta J(\theta), \ \ where \ \ J(\theta) = \frac{1}{N'} \sum_{n'=1}^{N'} r_\theta(\boldsymbol{x}'_{n'}) - \frac{1}{N} \sum_{n=1}^{N} r_\theta(\boldsymbol{x}_n)$$

この最適化問題を解いて得られる線形多項式 $r_\theta(\boldsymbol{x})$ を用いて、Kullback-Leibler 情報量は次式により近似できる[89]。

$$KL(P \parallel Q) \sim \frac{1}{n} \sum_{i=1}^{n} \log r(\boldsymbol{x}_i)$$

9.7.3 NIC の有効性評価

9.5 節の実験結果によって NIC 法が敵対的データを異常値として検出する手段として有効であることを示したが、有効であることの理由を説明するため、正常データと敵対的データの Kullback-Leibler 情報量を比較して、双方のデータの乖離度を推定した。まず、正常データと敵対的データ（攻撃手法：FGSM L^2）の画像データ 50 個（図 9.5 参照）に対する Kullback-Leibler 情報量を求めた結果を図 9.7 に示す。図 9.7 の Kullback-Leibler 情報量の近似値は 0.46 である。なお、図 9.4 に示したように、1 画像に対して複数の NIC が存在するが、図 9.7 では、各画像の複数の Kullback-Leibler 情報量の平均値を示している。

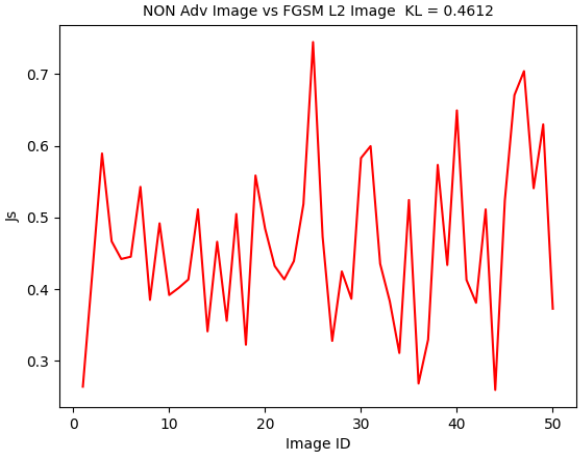


図 9.7 正常データと敵対的データの Kullback-Leibler 情報量

次に、正常データと敵対的データの NIC の Kullback-Leibler 情報量（図 9.5 の画像データから作成）を求めた結果を図 9.8 に示す。図 9.8 の Kullback-Leibler 情報量の近似値は 4.47 である。図 9.7 と図 9.8 の Kullback-Leibler 情報量には約 10 倍の違いがある。この違いが、正常データに加えられた摂動を NIC 法によってより集約された形で収集できることを表していると推定される。

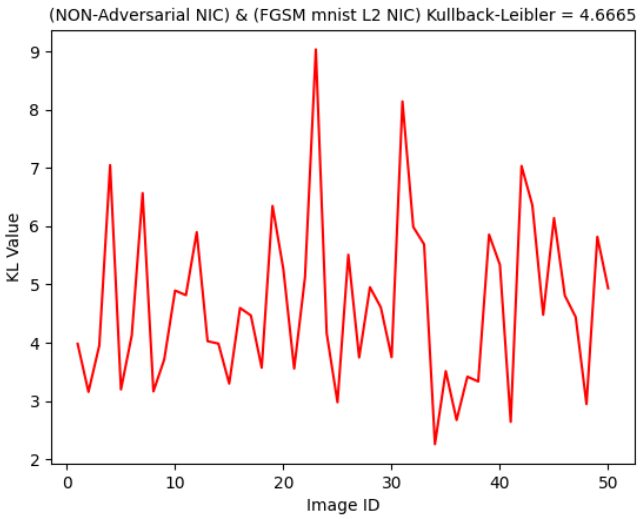


図 9.8　正常データと敵対的データの NIC の Kullback-Leibler 情報量

10 運用時における AI 品質管理技術

本章では、運用時における AI 品質管理技術として、コンセプトドリフトと呼ばれる時間経過に伴うデータ分布の変化に対し、その分布変化の検知と、変化後の分布に機械学習モデルを適応させる最新技術および同技術の関連技術である教師なしドメイン適応技術の最新技術に関する調査結果について報告する。

コンセプトドリフトは、運用中の AI システム内で稼働する機械学習モデルの性能低下を引き起こす主な原因の 1 つである。そのため、システムの運用開始時点で充足されていた品質を、運用期間を通じて維持するためには、ドリフトが生じているか否かを継続的に監視することに加え、必要に応じてシステム内の機械学習モデルを最新のデータを用いて再学習することで、ドリフト後の分布にシステムを適応させることが必要である。特に近年の機械学習技術の利用拡大に伴い、今後の AI システムの運用場面では、これまで扱われなかった種類のデータを含め、正解ラベル付けされていない大量のデータを短期間で処理することが求められる。

そこで、2019～2020 年度において、運用中の機械学習モデルの性能維持を目的として、上記のコンセプトドリフトの検知および適応を行う最新技術に関する調査を行った。その結果、これまでに開発されている手法の多くが、検知および適応時に運用中に新たに取得した入力データの正解ラベルを用いる教師あり手法であった。しかしながら、正解ラベルは必ずしも入手できるとは限らず、入手できたとしてもコストがかかる場合が多い。そのため、適用可能性を広げるため、または運用コストを削減するためには、それらの正解ラベルを用いない「教師なし手法」や、少数の正解ラベルのみ限定的に用いる「半教師あり手法」が有望であることがわかった。そこで、その視点から整理し検討した調査結果をサーベイとしてまとめた。各サーベイに関する詳細については、検知手法に関しては機械学習品質マネジメントガイドライン[1]7.8 節を、適応手法に関しては文献[90]をそれぞれ参照されたい。

一方、今後の AI システムの運用においては、新たに、データのプライバシやその可搬性の観点から適応前に使用した訓練データに依存しない適応技術や、入力データの分布以外の変化にも対応可能な適応技術の必要性も高まってきている。特に訓練データ（ソースデータ）に依存しない適応技術は、「ソースフリードメイン適応手法」や「test-time 適応手法」とも呼ばれ、ソースデータの管理や送受信に係る費用削減、同データの保管に関するセキュリティの観点からも注目を集めている。

そこで 2021 年度においては、上述の 2020 年度までの調査に引き続き、教師なしコンセプトドリフト適応技術および教師なしドメイン適応技術を中心に、2019 年以降の機械学習分野の主要国際会議で発表されたデータ変化に対する教師なし適応技術の最新の研究動向に関して調査を行った。その結果、近年においては、従来の教師なし適応技術に加え、上述のソースフリーな適応技術や、ラベルシフトなどの入力データの分布以外の変化にも対応可能な適応技術の開発が行われていることがわかった。さらには調査した論文に記載された幾つかの手法に関しては、画像分類問題だけでなく、セマンティックセグメンテーション問題や物体検知問題に対する有効性の検証も行われている。これら教師なし適応技術の研究動向は、データプライバシーの保持などの AI 運用における新たな課題に対して解決を図るとともに、今後の AI 運用における様々な場面においての活用へと広がっていると言える。本サーベイに関する詳細については、文献[91]を参照されたい。

11 参考文献リスト

（1 章の参考文献）

[1] 大岩 寛他, 機械学習品質マネジメントガイドライン 第 3 版, Digiarc-TR-2022-05, CPSEC-TR-2022006, 2022 年 8 月. https://www.digiarc.aist.go.jp/publication/aiqm/
[2] 宮城 優里, 大西 正輝, 作業者情報に注目した機械学習モデル比較可視化手法, 第 24 回画像の認識・理解シンポジウム, I31-22, 2021 年 7 月.
[3] 宮城 優里, 大西 正輝, 機械学習モデルの品質保証・評価のための作業者情報比較可視化手法, 第 49 回可視化情報シンポジウム, OS12, 2021 年 9 月.
[4] 高瀬朝海, 星野貴行, 畳み込みニューラルネットワークの特徴マップへの Data Augmentation 適用, 第 23 回画像の認識・理解シンポジウム,2020 年 8 月.
[5] Tomoumi Takase, Dynamic batch size tuning based on stopping criterion for neural network training, Neurocomputing, Volume 429, pp.1-11, 2021 年 3 月.
[6] 中島 震, 敵対的なセマンティック・ノイズの実行時検知, 情報処理学会・ソフトウェア工学研究会, 2020 年 7 月.
[7] 中島 震, 統計的な部分オラクルによるテスティング方法, 日本ソフトウェア科学会大会, 2020 年 9 月.
[8] 中島 震, ニューラルネットワーク・ソフトウェアの頑健性検査, 情報処理学会・ソフトウェア工学研究会, 2020 年 11 月.
[9] Shin Nakajima (NII), Software Testing with Statistical Partial Oracles, 10th SOFL+MSVL, 2021 年 3 月.
[10] 中島 震, 訓練済み機械学習モデル歪みの定量指標, 電子情報通信学会・ソフトウェアサイエンス研究会, 2021 年 3 月.
[11] 大川 佳寛, 小林 健一, ラベルなし運用データに対するコンセプトドリフト適応技術に関するサーベイ, 第 35 回 人工知能学会全国大会, 2021 年 6 月.
[12] 大川 佳寛, 小林 健一, データ変化に対する教師なし適応技術に関する最新研究動向とその考察, 第 36 回 人工知能学会全国大会, 2022 年 6 月.

（2 章の参考文献）

[13] 原 聡, 私のブックマーク「機械学習における解釈性」, 人工知能, vol. 33, no. 3, pp. 366-369, 2018.
[14] Fred Hohman, Minsuk Kahng, Robert Pienta, Duen Horng Chau, Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers, IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 8, pp. 2674-2693, 2018.
[15] Bilal Alsallakh, Amin Jourabloo, Mao Ye, Xiaoming Liu, Liu Ren, Do Convolutional Neural Networks Learn Class Hierarchy?, IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 152-162, 2018.

[16] Mengchen Liu, Jiaxin Shi, Kelei Cao, Jun Zhu, Shixia Liu, Analyzing the Training Processes of Deep Generative Models, IEEE Transactions on Visualization and Computer Graphics, vol.24, no.1, pp.77-87, 2018.
[17] Jorge Piazentin Ono, Sonia Castelo, Roque Lopez, Enrico Bertini, Juliana Freire, Claudio Silva, PipelineProfiler: A Visual Analytics Tool for the Exploration of AutoML Pipelines, IEEE Transactions on Visualization and Computer Graphics, vol.27, no.2, pp.390-400, 2021.
[18] Saleema Amershi, Maya Cakmak, W. Bradley Knox, Todd Kulesza, Power to the People: The Role of Humans in Interactive Machine Learning. AI Magazine, vol.35, no.4, pp.105-120, 2014.
[19] Heungseok Park, Jinwoong Kim, Minkyu Kim, Ji-Hoon Kim, Jaegul Choo, Jung-Woo Ha and Nako Sung, VISUALHYPERTUNER: VISUAL ANALYTICS FOR USER-DRIVEN HYPERPARAMTER TUNING OF DEEP NEURAL NETWORKS, 2019.

（4 章の参考文献）

[20] Gontijo-Lopes, R., Smullin, S. J., Cubuk, E. D., and Dyer, E., Affinity and Diversity: Quantifying Mechanisms of Data Augmentation. arXiv preprint arXiv:2002.08973, 2020.
[21] Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q., RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In Neural Information Processing Systems, 33, 2020.
[22] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D., Mixup: Beyond Empirical Risk Minimization. In International Conference on Learning Representations, 2018.
[23] Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y., Manifold Mixup: Better Representations by Interpolating Hidden States. In International Conference on Machine Learning, pp. 6438–6447, PMLR, 2019.
[24] Kim, J-H., Choo, W., and Song, H. O., Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In International Conference on Machine Learning, 2020.
[25] Beckham, C., Honari, S., Verma, V., Lamb, A., Ghadiri, F., Hjelm, R. D., Bengio, Y., and Pal, C. On adversarial mixup resynthesis. In Neural Information Processing Systems, 2019.

（5 章の参考文献）

[26] 中島 震, ソフトウェア工学から学ぶ機械学習の品質問題, 丸善出版 2020.
[27] Pei, K., et al., DeepXplore: Automated Whitebox Testing of Deep Learning Systems, In Proc. 26th SOSP, 2017, pp.1-18.
[28] Nakajima, S., Distortion and Faults in Machine Learning Software, In Post-Proc. 9th SOFL+MSVL, 2020, pp.29-41.
[29] Ma, L., et al., DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems, In Proc. ASE, 2018, pp.120-131.
[30] Tian, Y., et al., DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars, In Proc. 40th ICSE, 2018, pp.303-314.

[31] Zhang, M., et al., DeepRoad: GAN-Based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems, In Proc. ASE, 2018, pp.132-142.
[32] Zhang, P, et al., CAGFuzz: Coverage-Guided Adversarial Generative Fussing Testing of Deep Learning Systems, arXiv:1911.07931, 2019.
[33] Harel-Canada, F., et al., Is Neuron Coverage a Meaningful Measure for Testing Deep Neural Networks? In ESEC/FSE, 2020, pp.851-862.
[34] Kim, J. et al., Guiding Deep Learning System Testing Using Surprise Adequacy, In Proc. 41st ICSE, 2019, pp.1039-1049.

（6 章の参考文献）

[35] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, The MIT Press 2016.
[36] Simon Haykin, *Neural Networks and Learning Machines (3ed.)*, Pearson India 2016.
[37] Shiqing Ma, Yingqi Liu, Wen-Chuan Lee, Xiangyu Zhang, and Anath Grama, MODE: Automated Neural Network Model Debugging via State Differential Analysis and Input Selection, In Proc. 26th ESE/FSE, pp.175-186, 2018.
[38] Shin Nakajima, Software Testing with Statistical Partial Oracles – Applications to Neural Network Software, In Proc. 10th SOFL+MSVL, pp.275-192, 2021.
[39] Shin Nakajima and Tsong Yueh Chen, Generating Biased Dataset for Metamorphic Testing of Machine Learning Programs, In Proc. 31st ICTSS, pp.56-64, 2019.
[40] Gregor Montavon, Genevieve B. Orr, and Klaus-Robert Muller (eds.), *Neural Networks: Tricks of the Trade (2ed.)*, Springer 2012.
[41] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov, Membership Inference Attacks Against Machine Learning Models, arXiv:1610.05820v2, 2017.
[42] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha, Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting, arXiv:1709.01604v5, 2018.
[43] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen, Understanding Membership Inferences on Well-Generalized Learning Models, arXiv:1802.04489, 2018.
[44] Charu C. Aggarwal, *Outlier Analysis (2ed.)*, Springer 2017.
[45] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer, Replux: An Efficeint SMT Solver for Verifying Deep Neural Networks, In Proc. 29th CAV, pp.97-117, 2017.
[46] Pang Wei Koh and Percy Liang, Understanding Black-box Predictions via Influence Functions, arXiv:1703.04730v3, 2020.
[47] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana, DeepXplore: Automated Whitebox Testing of Deep Learning Systems, In Proc. 26th SOSP, pp.1-18, 2017.
[48] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, Jianjun Zhao, and Yadong Wang. DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems, In Proc. 33rd ASE, pp.120-131, 2018.
[49] Yizhen Dong, Peixin Zhang, Jingyi Wang, Shuang Liu, Jun Sun, Jianye Hao, Xinyu Wang, Li Wang, Jin Song Dong, and Dai Ting. There is Limited Correlation between Coverage and

Robustness for Deep Neural Networks. arXiv:1911.05904, 2019.

[50] Fabrice Harel-Canada, Lingxiao Wang, Muhammad Ali Gulzar, and Miryung Kim, In Proc. 28th ESEC/FSE, pp.851-862, 2020.

[51] Shin Nakajima, Distortion and Faults in Machine Learning Software, In Proc. 9th SOFL+MSVL, pp.29-41, 2019.

[52] Stephanie Abrecht, Maram Akila, Sujan Sai Gannamaneni, Konrad Groh, Christian Heinzemann, Sebastian Houben, and Matthjas Woehrle, Revisiting Neuron Coverage and Its Application to Test Generation, In Proc. SAFECOMP 2020 Workshop, pp.289-301, 2020.

[53] 『機械学習品質評価・向上技術に関する報告書（第 1 版）』第 4 章, DigiARC-TR-2021-01 and also CPSEC-TR-2021002, 産業技術総合研究 2021.

（7 章の参考文献）

[54] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, Intriguing properties of neural networks, The International Conference on Learning Representations (ICLR 2014), pp.1-10, 2014. https://arxiv.org/abs/1312.6199

[55] Guy Katz, Clark Barrett, David Dill, Kyle Julian, and Mykel Kochenderfer, Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, International Conference on Computer-Aided Verification (CAV), 2017. https://arxiv.org/abs/1702.01135

[56] Vincent Tjeng, Kai Xiao, and Russ Tedrake, Evaluating robustness of neural networks with mixed integer programming, International Conference on Learning Representations (ICLR), 2019. https://arxiv.org/abs/1711.07356

[57] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S. Dhillon, and Luca Daniel, Towards Fast Computation of Certified Robustness for ReLU Networks, International Conference on Machine Learning, PMLR 80, pp.5276-5285, 2018. https://arxiv.org/abs/1804.09699

[58] Akhilan Boopathy, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel, CNN-Cert: An Efficient Framework for Certifying Robustness of Convolutional Neural Networks, The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019), pp.3240-3247, 2019. https://arxiv.org/abs/1811.12395

[59] Tsui-Wei Weng, Pin-Yu Chen, Lam Nguyen, Mark Squillante, Akhilan Boopathy, Ivan Oseledets, and Luca Daniel, PROVEN: Verifying Robustness of Neural Networks with a Probabilistic Approach, International Conference on Machine Learning (ICML 2019), PMLR vol. 97, pp.6727-6736, 2019. http://proceedings.mlr.press/v97/weng19a.html

[60] Nicholas Carlini and David Wagner, Towards Evaluating the Robustness of Neural Networks, IEEE Symposium on Security and Privacy (SP), pp.39-57, 2017. https://arxiv.org/abs/1608.04644

[61] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel, Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach, International Conference on Learning Representations (ICLR 2018), 2018.

https://arxiv.org/abs/1801.10578

[62] Eric Wong and J. Zico Kolter, Provable defenses against adversarial examples via the convex outer adversarial polytope, International Conference on Machine Learning (ICML 2018), PMLR vol. 80, pp.5283-5292, 2018. https://arxiv.org/abs/1711.00851

[63] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana, Certified Robustness to Adversarial Examples with Differential Privacy, The IEEE Symposium on Security and Privacy (SP), 2019. https://arxiv.org/abs/1802.03471

[64] Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter, Certified Adversarial Robustness via Randomized Smoothing, The 36th International Conference on Machine Learning (ICML 2019), 2019. https://arxiv.org/abs/1902.02918

[65] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, Towards Deep Learning Models Resistant to Adversarial Attacks, The Sixth International Conference on Learning Representations (ICLR 2018), 2018. https://arxiv.org/abs/1706.06083

（8 章の参考文献）

[66] Guillermo Valle-Pérez and Ard A. Louis, Generalization bounds for deep learning, arXiv:2012.04115v2, 2020. https://arxiv.org/abs/2012.04115

[67] Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014. https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/index.html

[68] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio, Fantastic Generalization Measures and Where to Find Them, International Conference on Learning Representations (ICLR 2020). https://arxiv.org/abs/1912.02178

[69] Konstantinos Pitas, Mike Davies, and Pierre Vandergheynst, PAC-Bayesian Margin Bounds for Convolutional Neural Networks, arXiv:1801.00171, 2018. https://arxiv.org/abs/1801.00171

[70] Konstantinos Pitas, Dissecting Non-Vacuous Generalization Bounds based on the Mean-Field Approximation, ICML 2020. arXiv:1909.03009, 2020. https://arxiv.org/abs/1909.03009

[71] Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz, SGD Learns Over-parameterized Networks that Provably Generalize on Linearly Separable Data, ICLR 2018. https://openreview.net/forum?id=rJ33wwxRb

[72] Ilja Kuzborskij and Christoph H. Lampert, Data-Dependent Stability of Stochastic Gradient Descent, 2017. arXiv:1703.01678. https://arxiv.org/abs/1703.01678

[73] Guillermo Valle-Perez, Chico Q. Camargo, and Ard A. Louis, Deep learning generalizes because the parameter-function map is biased towards simple functions, ICLR 2019. arXiv:1805.08522. https://arxiv.org/abs/1805.08522

[74] Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M. Roy, In search of robust measures

of generalization, NeurIPS 2020. arXiv:2010.11924. https://arxiv.org/abs/2010.11924

[75] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P. Adams, and Peter Orbanz, Non-vacuous Generalization Bounds at the ImageNet Scale: a PAC-Bayesian Compression Approach, ICLR 2019. https://arxiv.org/abs/1804.05862

[76] Saurabh Garg, Sivaraman Balakrishnan, J. Zico Kolter, and Zachary C. Lipton, RATT: Leveraging Unlabeled Data to Guarantee Generalization, ICML 2021. arXiv:2105.00303. https://arxiv.org/abs/2105.00303

[77] Gintare Karolina Dziugaite and Daniel M. Roy, Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data, Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI), 2017. arXiv:1703.11008. https://arxiv.org/abs/1703.11008

[78] María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári, Tighter risk certificates for neural networks, Journal of Machine Learning Research, 2021. arXiv:2007.12911. https://arxiv.org/abs/2007.12911

（9 章の参考文献）

[79] X. Ma, Characterizing adversarial subspaces using Local Intrinsic Dimensionality, 2018.

[80] D. Meng , Magnet: a two-pronged defense against adversarial examples, in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017.

[81] W. Xu, Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks, in Proceedings of the 2018 Network and Distributed Systems Security Symposium (NDSS), 2018.

[82] Shiqing Ma, NIC: Detecting Adversarial Samples with Neural Network Invariant Checking, Network and Distributed Systems Security Symposium (NDSS), NDSS 2019.

[83] 産業技術総合研究所, AI Bridging Cloud Infrastructure, https://abci.ai/ja/

[84] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, vol. 86, no. 11, pp.2278–2324, 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[85] A. Krizhevsky and G. Hinton, Learning multiple layers of features from tiny images, 2009.

[86] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[87] Nicolas Papernot, Ian Goodfellow, Ryan Sheatsley, Reuben Feinman, and Patrick McDaniel. cleverhans v1.0.0: an adversarial machine learning library. arXiv preprint arXiv:1610.00768, 2016.

[88] CleverHans, https://github.com/cleverhans-lab/cleverhans

[89] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori, Density Ratio Estimation in Machine Learning, Cambridge University Press, 2012.

（10 章の参考文献）

[90] 大川 佳寛, 小林 健一, ラベルなし運用データに対するコンセプトドリフト適応技術に関するサーベイ, 第 35 回 人工知能学会全国大会，2021 年 6 月.
[91] 大川 佳寛, 小林 健一, データ変化に対する教師なし適応技術に関する最新研究動向とその考察, 第 36 回 人工知能学会全国大会，2022 年 6 月.