製薬協

機械学習始めました

日本製薬工業協会 医薬品評価委員会

データサイエンス部会 タスクフォース 1

Machine Learning

2020 年 5 月

目次

はじめに

日本製薬工業協会 医薬品評価委員会 データサイエンス部会では，2019 年 5 月に「AI ってなに？」[1]を公表し，機械学習及びそれによって作られる人工知能(Artificial Intelligence：AI)の基本的な仕組み，利用の可能性などを紹介した．また，それに先立って，製薬業界における AI の導入，利用状況のアンケート調査を行い，公表した[2]．その後，AI を取り巻く環境，特にクラウドコンピューティング（クラウド）の開発環境による初期投資の軽減と拡張の柔軟性，再利用可能な機械学習モデルを含むライブラリによる AI 開発コストの軽減や開発期間の短縮，複雑で経験を要した機械学習のチューニングも，自動化による技術的なハードル低下などによる”AI の民主化”が急速に進んでいる．これにより，データの理解・解釈やサービス分野の知識（ドメイン分野知識）はあるが，スキルや環境の不足により少し前までは，機械学習の利用が困難であった研究者，サービスの企画担当者及びデータサイエンティストにとって，機械学習は身近なツールとなりつつある．本報告書では，製薬企業に関わる広い意味でのデータサイエンティストがデータ解析や臨床データを含む身近なデータを用いたサービスの開発に機械学習及びそれによる AI を用いる場面を想定して，企画，データの加工，機械学習の実行について実際のデータを用いて紹介する．また，AI の開発で大きな障害となる学習データの量及び質の不足を補うデータを増やす方法及び，深層学習については，学習済みモデルを利用する方法を紹介する．深層学習をはじめから行うのはデータ確保の困難さから本書では扱わないこととした．

「第 1 章 機械学習を含むプロジェクト」では，機械学習を利用する場合に通常のシステム開発とは異なる点について概観する．「第 2 章 開発データの入手，加工（アノテーション)」及び「第 3 章 学習のためのデータ分割と学習方法」では，データの入手，増やし方，データの加工方法，機械学習のためのデータの分割まで，第 4 章及び第 5 章で用いている方法を含めて説明している．「第 4 章臨床試験データや RWD に機械学習を活用する実装例」では，インスリンを使用する糖尿病患者の血糖値などのデータを用いて，実際に機械学習を行い，サービスに用いるまでを詳細に紹介している．そして「第 5 章 学習済みモデルの利用に関する解説・実用例の紹介」では学習済モデルを改修して流用する転移学習及びファインチューニングの概念と実際を画像及び自然言語の事例を用いて紹介している．

本書で扱っている機械学習の事例

第 4 章	糖尿病患者の低血糖イベント予測
第 4 章第 7 節	深層学習による血糖値の予測
第 5 章第 2 節第 1 項 B)	PubMed-w2v モデルを用いた類似語抽出
第 5 章第 2 節第 2 項	MeCab，GiNZA を用いた形態素解析
第 5 章第 4 節	VGG19 のファインチューニングによるほくろと皮膚がんの識別
第 5 章第 5 節第 1 項	理化学研究所の高精度緑内障自動診断装置
第 5 章第 5 節第 2 項	電子的診療記録からの疾患診断コードの自動付与

R，Python のコードは巻末 Appendix を参照．

第１章　機械学習を含むプロジェクト

近年，大量のデータの蓄積，及びコンピュータ性能の向上により，機械学習を活用した製品・サービスが増えてきており，今や，機械学習を用いたサービスに触れたことのない人はいないと言っても過言ではない状況である．身近な例を挙げれば，Google 検索や Amazon のレコメンド機能，Siri の音声認識などにも機械学習が活用されている．このような動きは医療業界にも波及してきており，医薬品の研究開発分野においても活用事例が出てきている．

サービスの普及に伴い，個人情報の扱いを含めて法や倫理の問題についても議論されるようになってきた．また，機械学習の精度向上には大量のデータが欠かせないが，そのデータの帰属をめぐる問題や，意図しない思わぬ結果を導き出してしまうという問題も存在する．したがって機械学習をシステムに組み込む場合はこれらの問題についても念頭に置いて対処する必要がある．

機械学習とは，人工知能のプログラム自身が学習する仕組みであり，コンピュータは与えられたサンプルデータを通してデータに潜むパターンを学習するものである[3]．機械学習が活用される場面で，最も多いのが「識別と予測」であるが，データ量を増やす又はアルゴリズムを改善したとしても，その精度が 100%になることはない．機械学習の手法は，教師あり学習，教師なし学習，強化学習，の 3 つに大別することができる．教師あり学習は入力データと対になる正解からなるデータセットをあらかじめ準備して行う機械学習を指し，判別と回帰の分野に活用されている．教師なし学習は正解データが用意されてないデータセットに対して，機械自らがその特徴や構造を分析するものであり，グループ分けや情報の要約に活用されている．強化学習は，一連の行動系列の結果としての報酬を最大とするように学習を行うものであり，将棋や囲碁のようなゲームなどに使われている．

上記のような特徴から，機械学習はどんな分野にも応用できるものではなく，以下のようなデメリットも指摘されている．

- そもそも予測が困難な課題には対応できない[4]：
  - 偶発的に起こる事象
  - 現象が起きるメカニズムが複雑または現象を説明するデータが十分に取得できないもの
  - 過去のデータがないもの
- 機械学習を含むシステム（以下「AI システム」）構築が難しい[5]：
  - 確率的な処理があるためシステムの動作テストの自動化に向かない
  - 長期運用しているとトレンドの変化などで入力（データ）の傾向が変化する
  - データ処理，システム動作が複雑になる
  - データの依存関係が複雑になる

- 実験コードやパラメータが残りやく，バグ，セキュリティホールにつながりやすい
- 開発と本番のプログラミング言語／フレームワークが異なりやすい

機械学習の手法の利用は，学習により得られたモデルをサービス，システムに組み込む場合と，データ解析として用いるのみで，再利用しない場合がある．本章では前者について紹介する．

第 1 節 機械学習プロジェクトの進め方

AI システム開発[注 1]の方法論として，機械学習工学が提唱されてきているが[6]，現時点において，機械学習プロジェクトを進める上での明確な方法論は存在しない．しかしながら，様々な機械学習プロジェクトの実績が積み重なっていく中で，特有のステップや要点が示されている．本章ではこれらを踏まえて機械学習プロジェクトに必要なステップを概観する．機械学習プロジェクトを進めるには，構想，PoC（Proof of Concept：概念実証）[注 2]，実装，運用の 4 つのステップに大別することができるが[4]，その境界は必ずしも明確ではない．既存の機械学習を用いないシステム開発プロジェクトと大きく異なる点としては，PoC の存在が挙げられる．また，PoC においては試行錯誤を繰り返すことから，開発中に仕様の検討，変更を行うアジャイル型で進めることが多くなる．従来の予め仕様を決るウォーターフォール型と比較したメリットとデメリットを表 1 -1 にまとめた．

表 1 -1 ウォーターフォール/アジャイル型のメリットとデメリット[3]

	ウォーターフォール型	アジャイル型
概要	システムの開発を「要件定義」「基本計画」「外部（概要）設計」「内部（詳細）設計」「プログラム設計」「プログラミング」「テスト」という工程に分けて Step by Step で行う方法．原則として後戻りしない．この場合でも，アジャイル同様にテストは機能ごとの単体テストと全体の結合テストが行われる．	システム開発中に仕様，設計の変更があることを前提に，初めに厳密な仕様設定を行わず，要件から必要な小さな機能に分けて，機能ごとにおよその仕様でイテレーションと呼ばれれる短期間（1 から 4 週間など）で設計-開発-単体テストを繰り返して徐々に開発を進め，結合テストを行う方法．
メリット	・責任範囲が明確 ・プロジェクト管理が容易	・開発着手時点で，ゴールを厳密に定義できない開発に適する ・仕様変更に柔軟に対応可能
デメリット	・試験段階で不備があれば，後戻りで大幅な時間ロスがありえる（完成時には市場ニーズから乖離していることも） ・試行錯誤が必要な開発に適さない	・全工程にユーザが関与するため責任範囲や成果の帰属が不明確となり得る ・プロジェクト管理が難しい

注1 本章は，主にシステム，サービスに組込んで繰り返し使用する AI を含む開発が対象であり，データ解析の手段または短期間の利用を目的とする場合は，多くの工程を省略可能である．

注2 機械学習のプロジェクトは創薬と似ており，そのプロジェクトの実現可能性，効用・効果，用いることができるデータ，機械学習の技術的な観点から検証するプロセスが必要となる．

第 1 項 プロジェクトに必要なリソース

機械学習プロジェクトにおいて必要なリソース（人材，物資，資金，データ）が社内で確保できない場合は，パートナーを探す必要がある．

機械学習プロジェクトにおいて必要とされる人材は，サービスや製品の知識を含めてビジネスニーズを理解している人材とプログラムを開発するエンジニアの他に，データサイエンスに明るい人材（データサイエンティスト）が不可欠である．これは既存の他のシステム開発プロジェクトとは異なる点である．有賀らは，機械学習を含めたサービスや製品をビジネスとして成功させる上では

①サービスや製品に関するドメイン知識（学習に用いるデータの意味を理解しており，説明及び解釈ができる）を持った人，

②統計や機械学習に明るい人，

③データ分析基盤を作れるエンジニア能力のある人，

④失敗しても構わないとリスクを取ってくれる責任者

が必要だとしている[5]．現時点では機械学習にも対応可能なデータサイエンティストと呼べるような人材は非常に限られており，いずれの企業においても，そのような機械学習を含むプロジェクトで即戦力となる人材を確保することは未だ困難なようである．したがって，社内で育成するか外部ベンダーに協力を仰ぐ必要が出てくる．但し，戦略論等でも繰り返し述べられることではあるが，アウトソースすると，その業務のノウハウが社内で蓄積できなくなってしまうため，長期的に中核事業に育てる場合は，社内に人材を確保することが望ましいであろう．

機械学習において必要とされる物資は，十分な計算能力のあるハードウェアと，目的に適した開発環境ソフトウェア[注 3]である．代表的なソフトウェアとして Anaconda（言語は Python）や Rstudio（言語は R）がある．また，機械学習のためのプログラムコードやデータセットなどが GitHub（https://github.com/）といったサイトに公開されており，これらを利用することにより効率的に開発を進めることもできる（詳細は第 2 章第 3 節第 1 項表 2 -4 参照）．

データは機械学習を進める上で非常に重要な要素であり，機械学習を用いる目的（推論や分類）を達成可能なデータが活用可能な形で社内に存在するかを考える必要がある．社内にデータがない場合は，非臨床・臨床試験や調査などを通じて取得することや，外部のデータ購入を検討することとなる．社内外いずれの場合においても，使用するデータの特性や限界をよく理解した上でプロジェクトを進めることが肝要であり，これらを考慮せずに開発した AI システムでは求める精度に到達することは難しい．外部データを利用する際に気を付けなければならないことは，データの利用条件である．(1) 著作権法，(2) 不正競争防止法，

注3 ここでいう「ソフトウエア」は，プログラミング言語とそのコンパイラ/実行環境/グラフィカルインターフェース（GUI），各種ライブラリ，解析に用いるデータセットなどを包括して指す．

（3）個人情報保護法等，（4）個別の契約，（5）そのほかの理由により，データの利用に制約がかかっている場合がある．機械学習の成果を外部に出さなくても問題になるケースもあるため，注意が必要である[3]．経済産業省から 2018 年の不正競争防止法改正（2019 年 7 月施行）による「限定提供データ」の不正取得や使用等に関する民事措置及び，これに先立つ同年 1 月に公表された「限定提供データに関する指針」を踏まえて改正された「AI・データの利用に関する契約ガイドライン 1.1 版」[7]が公表されているので，参考にされたい．

外部のリソースを活用する際の参考として，韮原は以下のようなフローチャート[4]を提示しており，プロジェクトを進める上では参考にされたい．

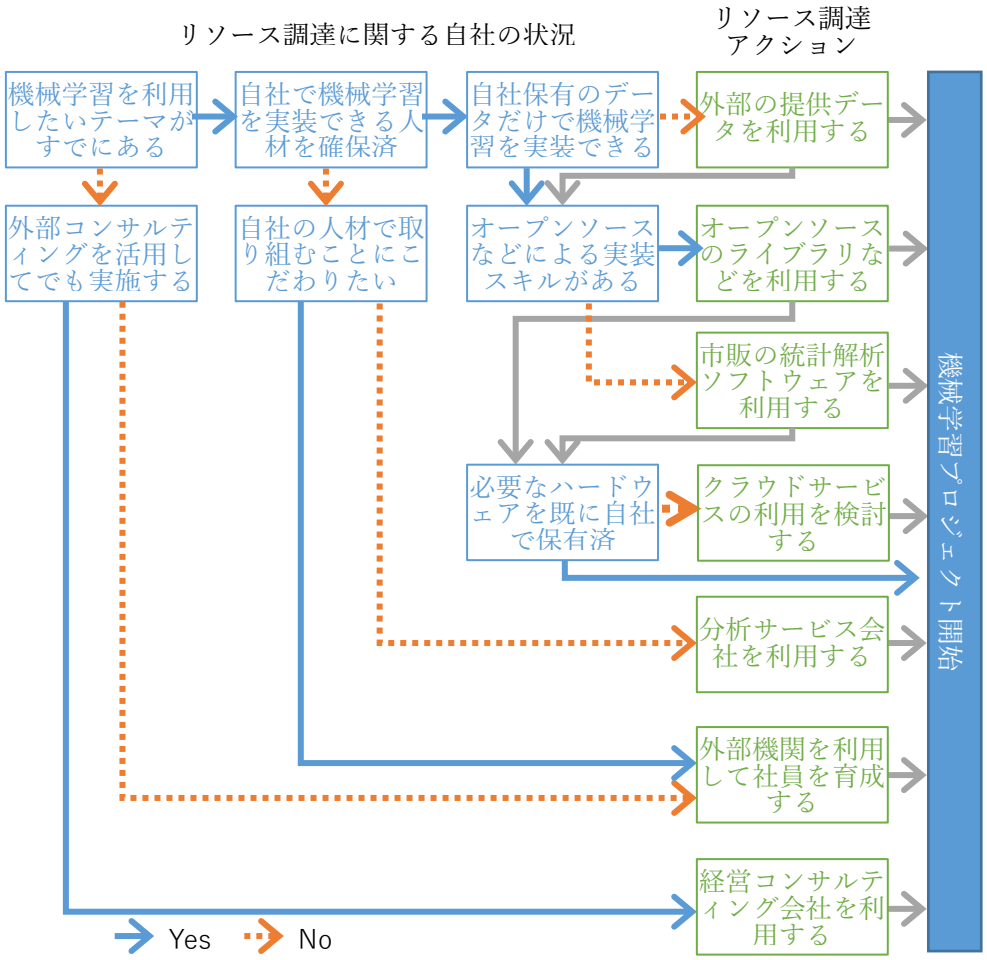


図 1 -1 機械学習に取り組む際のリソース調達先（書籍[4]を一部改変）

第 2 項 構想フェーズ

構想フェーズでは，解決すべき課題を選定し，それを実行計画に落とし込んだ上で，チームメンバー及びステークホルダーとの合意を得るところまでが含まれる．課題を選定する際には，効果を客観的に測定するために，より具体的でアクション可能なレベルまでブレイ

クダウンして考え，問題を定式化する[5]．続いて，機械学習を用いる必要があるか否かを検討する．機械学習プロジェクトは，数あるソフトウェア開発プロジェクトの中でも，極めて不確実性の高い開発プロジェクトと言える．一般的なソフトウェアは，事前の設計の段階である程度，最終的なソフトウェアのアウトプットを定義できる一方，機械学習プロジェクトでは試行錯誤を重ねないと課題解決に至る方法にたどり着けない場合がほとんどだと言われている[8]．更に 2 ページ後半でも触れているが，AI システムは通常のシステムより技術的負債が蓄積しやすい[注 4] [5]ため，安易に機械学習を選択するのではなく，そのほかの手段で解決が困難な課題について機械学習を用いるようにすべきである．有賀らは，大量のデータに対して高速に安定して判断を求める必要がある，予測結果には一定数の間違いが含まれることが許容できる，の 2 つの条件を満たす場合機械学習を用いるとしている[5]．

また，具体的なアクションに落とし込む際には，AI システム全体として考えることが重要となる．機械学習は単体で機能するものではないため，機械学習を用いてどの業務を効率化し，プロセスとしての前後の関係はどのようになるのかなど，業務の流れ全体をデザインする必要がある[4]．AI システム設計上のポイントは，予測結果をどういう形で利用するのか，予測が誤っていた場合，それをどこで吸収するのか，の 2 点である．機械学習に 100%正解を出力するアルゴリズムは存在しないことから，AI システム全体でどのように誤りをカバーするのか，人手による確認や修正は必要なのか，必要だとしたらどこでそれを実現するのかを考えることも AI システム設計に含まれる[5]．

第 3 項 PoC フェーズ

AI システム開発の計画時においては，最終的に得られる精度がどの程度になるかを正確に予測するのは困難であることから，技術的な実行可能性を評価する PoC のステップが不可避となる．PoC フェーズにおいては，①アルゴリズムの選定，②学習及び教師データの用意，③データの前処理（特徴抽出・加工含む），④学習・パラメータチューニングを通じてモックアップモデルを構築し[5]（詳細は 3 章以降参照），事前に設定した項目の評価・検証を行うことで機械学習のアルゴリズム，得られるモデル評価する．評価の結果，性能が得られない原因がデータの量又は質の不足が原因である場合には，必要に応じてこのステップを探索的に繰り返し，目標とする精度が得られるか確認する．また，必要に応じて外部のデータを新たに取得することで精度の改善を図ることもある（外部からの情報取得の際の注意点は第 2 節第 2 項を参照）．これと同時に，構想フェーズのシステム設計に基づいて実装することで業務やサービスを運営した場合，無理がない AI システムが構築できるかというオペレーションの観点からも検証する必要がある．また，実行スケジュール上やビジネス上

注4 技術的負債とは，設計，プログラムコードが理想的な状態から離れていること．具体的には，複雑，冗長，コーディング規約に従っていない，廃止予定の警告を無視しているなどのプログラムコード，古いバージョンのライブラリの利用，開発環境と運用環境の相違，運用負荷・負担の過小な見積もり，実装内容のブラックボックス化など．AI システムの開発は仕様変更が生じやすい，データなど環境の影響が大きいなどにより，負債が蓄積しやすい．

の制約を満たすことができるかも合わせて確認することになる．これらの項目がすべてクリアしていれば，実装フェーズに進むことになるが，なかなか構想通りにうまくいくことばかりではない．そのような場合にはクリアできない部分の対応策も含めて検討し，次のフェーズに進むか中止するか判断することになる[4]．

第 4 項 実装フェーズ

PoC フェーズの次は AI システムを実装するフェーズである．実装フェーズで必要な作業自体は通常のシステム開発と同様であるが，AI システムの根幹をなす機械学習モデル如何によって必要な実装が定まるため，進め方や留意点には大きな違いがある．つまり，要件定義，設計，開発，テストという流れは変わらないが，設計に入る前に，用いる機械学習モデルで使用するデータ，前処理，アルゴリズム，出力するための処理，性能の監視及び改善のためのログの設計などの機能要件を確定させておくことが，設計以降の工程において手戻りを発生させないために重要となる．また，非機能要件（可用性，性能・拡張性，運用・保守性，移行性，セキュリティ，システム環境・エコロジー[注 5]）についても必要な事項を要件として定義しておくことになる．

AI システムを設計する際には，システム開発者以外のユーザにもわかるレベルで記載される「基本設計」と，開発者が実際のプログラムに落とし込む際に齟齬が出ないようにする「詳細設計」に分けて設計書を作成する．AI システムにおいては，アルゴリズムが読み込むデータの形式，前処理の自動化，モデルからの出力結果の形式などデータの扱いには特に気を配って綿密な設計を行う必要がある[4]．

第 5 項 運用フェーズ

機械学習プロジェクトに限ったことではないが，システムは実装して終了となる訳ではない．特に，AI システムは，モデルを構築するために使用したデータと現実との乖離が起きた場合に著しく精度が低下する可能性を孕んでいるため，運用時のデータと結果のモニタリングや現実の変化に合わせたチューニングが必要になる．そのためには適切な重要業績評価指標（key performance indicator：KPI）とログを設定し，これらを注意深くモニタリングしていくことになる．当然のことながら，KPI とログは計画と異なった場合に，原因を特定でき，改善のアクションを講じることができる粒度まで落とし込んだものを設定することが肝要である．KPI は，①ビジネス上の成果に関連するもの，②機械学習モデルの精度に関連するもの，③AI システムの稼働状況（安定性を含むシステムの品質）に関するもの，と 3 種類について設定する[4]．機械学習のデメリットとして先に挙げているが，機械学習によるモデルを長期運用していると，新しい治療薬の市場投入や治療ガイドラインの大幅な変更など，入力の傾向が変わることがある．これにより，予測性能が徐々に，あるいは

注5 「システム環境・エコロジー」とは，システムの非機能要件の 1 つで，システムの設置環境や消費エネルギー，CO2 排出量，環境負荷などエコロジーに関する事項．

急激に劣化する場合がよくある[5]．このような場合は，既存の機械学習モデルが現実世界の変化に対応しきれていないということであり，モデルに用いた変数の重みやハイパーパラメータ[注6]の再調整，変数自体の取捨選択，学習データ量・取得時期の変更，前処理の調整などの対応を取ることになる．この他にも，更に精度を高めたいという需要がある場合はチューニングを行うこともある[4]．

第２節 法と倫理について

機械学習モデルを応用したサービスや製品が身の回りに増えるにしたがい，製品の精度だけでなく，利用したモデルやその結果に関する著作権やプライバシーの侵害に関わる法的問題や，特定の集団にとって不利益となるような結果を導き出してしまうという倫理的問題などが取り上げられるようになってきている．そこで，本節においては，機械学習モデルの利用に関する法的・倫理的問題に焦点を当てて紹介することとする．

第１項 機械学習と倫理の問題

2015 年に Google Photo がアフリカ系の男女の写真に対して「ゴリラ」のラベルを付与してしまった問題は，機械学習モデルの予期せぬ結果の一例である．この例では，Google 社は謝罪し，解決を約束したが，2 年後でも，検索タグからゴリラやそのほかの霊長類の名前を削除することで対処されていた[9]．AI システムを利用する際には，特定の集団に対する偏見の助長につながる可能性や，プライバシー等への配慮の欠けた結果を導く恐れがあることを認識しておく必要がある．このような不測の事態を避けるためには，機械学習モデルを構築する際のデータには様々なバイアスが存在することを認識し，それが社会的に許容されるものなのか，許容されないとしたらどのような対策を講じる必要があるかという点について，注意深く評価するよう心掛けるべきである．そのためには，様々な意見を持つステークホルダーを巻き込んでプロジェクトを進めることが必要であり，それにより不測の事態を減らすことができると考えられる．一般的に利用可能なデータにおいては，学習データとなるサンプルが母集団を正しく反映していることは稀である．たとえ母集団を正しく反映していたとしても，例えば，COVID-19 流行前後では生活や購買が変容するように，その母集団自体の変化から偏りが生じる恐れがあり，結果的にこのようなサンプルから作られたモデルにおいては母集団のバイアスは維持されてしまう．また，データがそもそもデータベースに登録されていないため学習データに偏りが見られる場合もあることが指摘されている[3]．リアルワールドデータ（real-world data ：RWD）を使用する際には，こういった問題が頻発することが予想されることから，特に注意が必要な点である．

注6 機械学習（モデル）において，学習によって得られない（学習に自動的に最適化されない），人による調整・設定が必要なパラメータ．

第２項 学習済みモデル利活用の際の法的な留意点

大量の学習データを用いて学習を行い，特定の機能を持たせて利用可能な状態であるものが学習済みモデルである[10]．企業や大学又は研究機関で開発された一部の学習済みモデルが公開・共有されている．公開・共有された学習済みモデルを流用し，更なる派生モデルが開発されることで，技術開発の循環が起こっており，今後は，学習データを含む学習済みモデルの公開や共有の更なる進展が予想され，学習済みモデルは競争優位性を得るための差別化領域ではなくなるなど競争構造が変わる可能性を秘めている[11]．

クラウドベンダーが提供しているプラットフォームや GitHub（https://github.com/）というプログラムのコードやデータセットなどが管理されているサイトから学習済みモデルをオープンソースライセンスで利用することができる．しかしながら，修正 BSD ライセンス，MIT ライセンス及び Apache v2.0 ライセンスのようにコピーレフト条項[注 7]を持たず，著作権表示や免責事項の表示のみで，改変したものを商用ライセンスなど異なるライセンスに変更して公開することが可能なものから，GPLv2.0（GNU General Public license version 2.0）や LGPL v2.1（GNU Lesser General Public License version2.1）といった適用範囲の広いコピーレフト条項を備え，ライセンスを使用する著作物に基づき，派生的著作物にもコピーレフト条項の承継を求め，配布する対象への実質的な制限を含むものまである[12,13]．様々なライセンス形態があるため，学習済みモデルを利用する際には，著作権や商用目的への制限といった利用規約に注意を払う必要性がある．

本邦においても学習済みモデルの活用，保護など新たな知的財産上のあり方及び利活用の促進について議論・検討[10,14]が行われている．その中で一から新たに作成した場合であっても，学習済みモデルは，現行の知的財産制度上では「プログラムの著作物」や「プログラム等」として保護されうるとされる一方で，元のモデルとの関連性を特定することが技術的に困難であることを背景に現行の知的財産制度上の保護が不十分であるとの指摘もある[10]．こうした状況の中で，学習済みモデルの利用であっても，プログラムとして「発明」に該当するとして特許による保護の対象となった事例[15]もある．このようなことから，社内にて開発する場合は，ライセンスや知的財産の面に配慮しながら，手持ちのデータを用いて予備的に分析・評価し，目的を達成するために適切な学習済みモデルであるかを確認する必要がある．

一方で，社内のみでは目的を達成することが困難な場合は，外部委託といった選択肢もある．この場合は，外部委託先が作成した成果物が学習済みモデルとして納品されることになる．外部委託を行う際に，経済産業省の策定による「AI・データの利用に関する契約ガイドライン[7]」が参考になる．

実務上，学習済みモデルは，利用する者によって，学習済みパラメータを組み込んだ推論プログラムだけでなく，学習データセット，学習用プログラム，学習済みパラメータ及びそ

注7 ソフトウエアとともにソースコードを公開し，改変，複製，再配布を自由に認めるが，それらについても，元の著作権が保持され，すべての者が著作物を利用・再配布・改変できなければならないことが要求される．

のほかの派生的な成果物を含む概念として多義的に用いられる場合があり，確立した定義がないのが実情である．また，学習済みモデルは，従来型のソフトウェア開発と異なる特徴を有するため，次のような点に留意する必要性もある．

- 学習済みモデルの内容・性能等が契約締結時に不明瞭な場合が多い
  - 事前の性能保証が性質上困難であること
  - 事後的な検証等が困難であること
  - 探索的なアプローチが望ましいこと
- 学習済みモデルの内容，性能等が学習データセットに左右されること
- ノウハウの重要性が特に高いこと
- 生成物に更なる再利用の需要が存在すること

更に，学習済みモデルなどの成果物や学習データ及び学習済みパラメータの中には，知的財産権の対象となる又は対象にならないものが含まれるため，「権利帰属」や「利用条件」を定める必要がある．ベンダーとしては，これら学習済みモデルを含む生成物を再利用することで，新たな技術開発や事業展開の基礎となる要望があるのに対して，ユーザ側では，多大な費用と労力に基づき生成された成果物の再利用を制限したいとの意向も有ることから，利害調整の必要も生じる．このため，学習済みモデルの具体的な定義や範囲といった内容について，事前に契約の当事者間で十分に議論を行い，明確に定めておくことが望まれる[注8]．

注8 経済産業省「AI・データの利用に関する契約ガイドライン」[15]AI 編の「第 6.2 学習済みモデルの生成・利用で問題となり得る事項」にデータの取得，生成，著作権，学習済モデルの開発，権利帰属，利用条件などが説明されている．

第２章　開発データの入手，加工（アノテーション）

データは様々であり，個別の具体的な例示は難しいため，この章では簡単に概略を紹介する．具体的なデータの確認，加工などは，第３章を含め，第４章の糖尿病患者の低血糖イベントデータを用いて説明している．データの詳細は第４章第２節を参照いただきたい．

第１節 どのような，またどのくらいの量の学習データが必要か

学習に必要なデータ量は，データの種類と用いる手法，変数の数と性質により異なる．少ない学習データで高い精度が得られても新しいデータでは精度が得られない，学習データが大量にあっても似通ったデータが多く，多様性がない場合もやはり新しいデータでは精度が得られないといった過学習が生じる（学習に用いていない未知のデータへの対応能力を「汎化性能」と呼ぶ）．また，これは学習にバイアスを生じさせ，学習データの少ないグループのデータに対して誤った識別，予測が増加する．一般的に変数の数[注9]が多ければ必要な学習データも多く，多様性も必要となる．”It’s not who has the best algorithm that wins. It’s who has the most data.“（勝者は最も良いアルゴリズムを持つものではなく，最も多くのデータを持っているものだ．）[16] と言われるほど，データ量は重要であり，深層学習では，数万から数百万件のデータを用いられることは珍しくない．（表２-1）[17]

表２-1 AI プロジェクトで使用されたデータセットの推定規模

機械学習プロジェクト	タスク	データ数
FaceNet	顔検出と顔認識	45 万サンプル
MIT CSAIL	画像アノテーション	18 万 5 千件の画像，6 万 2 千件のアノテーション付き画像，65 万件のラベル付き物体
Sprout	Twitter の感情分析	数万件のツイート
Twitter Sentiment Analysis: The Good, the Bad and the OMG!	Twitter の感情分析調査	全 60 万データポイントのコーパスからのセレクション
Analysis and Classification of Arabic Newspapers' Facebook Pages using Text Mining Techniques	アラビア語 Facebook ページの感情分析と分類	6 万 2 千投稿，9 千コメント
Improved Text Language Identification for the South African Languages	テキスト言語識別	一言語につき 3 千件の学習サンプルと千件の試験サンプル
TransPerfect	機械翻訳	400 万単語
Building Chatbots from Forum Data: Model Selection Using Question Answering Metrics	チャットボット学習	20 万の質問と対になる 200 万の回答
Online Learning Library	自然言語処理研究	1 万 5 千学習ポイント，100 万以上の機能

注9 連続変数であれば，データテーブルの列の数と同じであるが，クラス・カテゴリカル変数の場合は個々のクラス・カテゴリ数－１が変数（ダミー変数）の数となる．

しかしながら，データの収集と非構造化データの場合は加工にも大変な時間と労力を要するため，まずは PoC を実施して必要なデータ量と質を見積ることや，ある程度学習モデルの開発が進んでいるならば，新たなデータの追加が学習モデルの精度を向上することに寄与するかを評価することが推奨される．例えば，高バイアス（正解から偏っている）なモデル[注 10]に新しいデータを追加しても，大きな成果は得らないことが多いため，データを収集するよりもモデルや特徴量の見直しを行う方が望ましい．逆に低バイアス・高バリアンス（正解から偏ってはいないがばらつきが大きい）なモデル[注 11]に対して，データを増やすことは有用であるため，新たなデータの取得にコストをかけることは合理的である．学習モデルのデータ量に対する評価方法として，学習曲線[注 12]がよく用いられる．（図 2 -1,図 2 -2, train，test の実線は実測データによる，破線は外挿による予測）[18]

また，分類の場合，対象の頻度が偏る（例えば陽性/陰性の二値で一方が 10%など）不均衡データ（Imbalanced Data)の場合は偏りがない場合（例えば陽性/陰性が半々の場合）に比べてより多くのデータ量が必要となる．また，データ量はモデルの精度に影響するため，必要とする精度にもよる．実際のところ，必要なデータ量は，実データを用いて試さないとわからない．

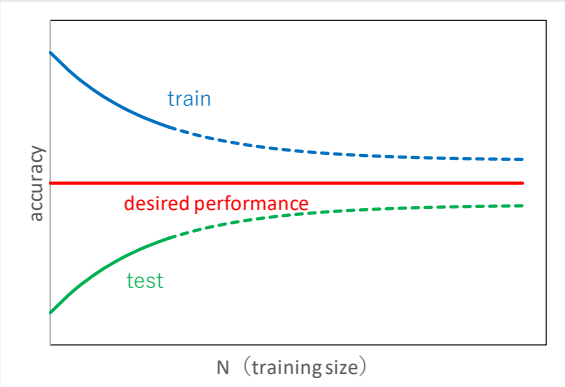


図 2 -1 高バイアスなモデルの学習曲線
（学習データを増やしても性能の向上は望めない）

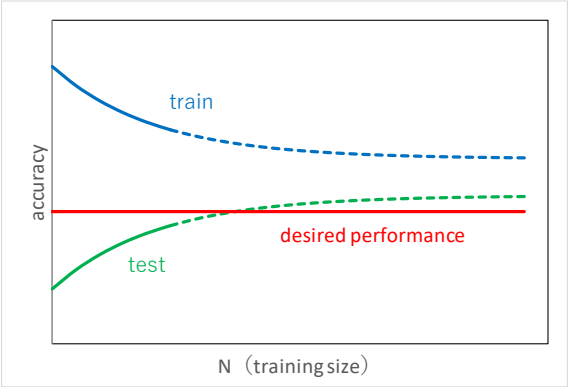


図 2 -2 高バリアンスなモデルの学習曲線
（学習データを増やすことで性能の向上が望める）

例示として第 4 章で用いるインスリン使用患者の低血糖イベントを予測するシミュレーションを行う．元のデータは 70 名の患者が含まれるが，低血糖イベント数が 15 回以上の 8 名のみを抽出し，更に低血糖イベント無しのデータを 50%除いたものを用いる（表 2 -2）．これはすでに説明したように，分類対象の偏りを小さくして学習しやすくするためである．

注10 学習不足で，実データをうまく表現できていないモデル．例えば，曲線を描くデータに対して，直線のモデルを採用しているようなケースでは，どれだけデータを増やしてもうまく学習できない．

注11 過学習しており，学習データに対して過度に学習しているモデル．学習データにはよく当てはまるモデルだが，汎用性がなく，他のデータには当てはまらない状態．

注12 横軸は学習のデータ量，縦軸は学習の評価指標．データが少ない場合，予測モデルは学習データを丸暗記できるしまう場合があるため，学習データに対して高い予測性能を示す．しかし，丸暗記をしたため，未知のデータであるテストデータに対して予測性能が低くなる．

このデータ（912 レコード）の 70%を学習用（638），残りを検証用として，学習データから 10%（64），20%（128），40%（255），60%（383），80%（510），100%（638）を各々 25 回抽出して基本的手法であるランダムフォレスト（random forest：rf）を用いて学習した結果の精度（accuracy），再現率（recall），特異度（specificity）を図 2 -3 と表 2 -3 に示す[注 13]．accuracy，specificity ともに 255 レコードで頭打ちとなっている．recall が 638 レコードでは逆に低下しているのは 25 回の繰り返しがすべて同じデータによるためである．25 回のシミュレーションのばらつきはレコード数が多くなるにしたがって小さくなっている．図 2 -3 には同じデータで勾配ブースティングマシーン（gradient boosting machine ：gbm）により学習した場合も示している．こちらはデータ量が少ない場合不安定な結果となっており，recall，specificity はデータ量の増加による逆転が生じている．このように，データ量の学習結果（精度）への影響は用いる指標，及び手法（アルゴリズム）によっても異なる．なお，この例は変数の少ない構造化データにシンプルな手法を用いているため，比較的少ないデータで性能が頭打ちとなっているが，より複雑なデータや，画像など非構造化データを用いる深層学習に分類される手法の場合に性能を向上させるためには，指数関数的にデータを増やす必要がある場合が少なくない．

反対に，主に学習の効率を向上するために，データ量を減らす場合がある．一つは，深層学習やそれ以外の機械学習で複雑なパラメータチューニングを行う，複数の手法を比較するなど，学習に多くの計算リソースと時間を要する場合がある．こういった場合，例えばある一定時間で学習できるデータ量に絞るなどが行われる．他には，学習データの質が低い，あるいは多様性が過ぎる場合に，学習の効率向上を目的として，学習しやすいデータに絞り込むことが行われる．

表 2 -2 抽出したデータの件数

	抽出データ		学習データ		検証データ	
	低血糖イベント		低血糖イベント		低血糖イベント	
患者ID	なし	あり	なし	あり	なし	あり
1	116	64	70	44	46	20
11	18	23	11	15	7	8
12	31	29	22	23	9	6
13	37	33	28	21	9	12
15	34	26	28	20	6	6
35	50	31	36	26	14	5
65	192	28	132	20	60	8
67	171	29	123	19	48	10
合計	649	263	450	188	199	75

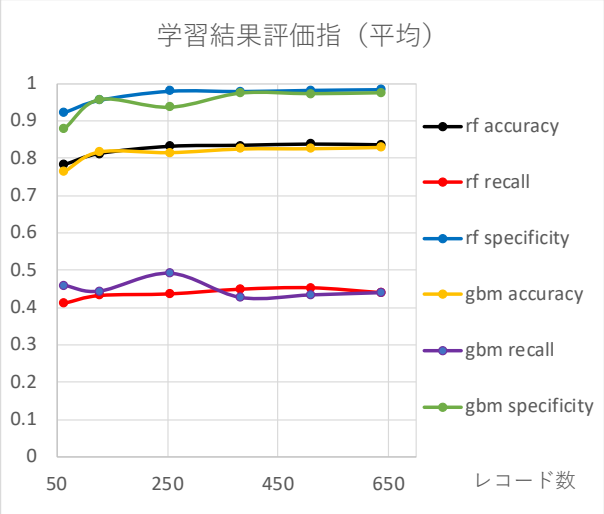


図 2 -3 学習結果の各指標の平均の推移

注13 レコード数は臨床試験などで得られやすい現実的な件数として選んでいる．精度（accuracy），再現率（recall），特異度（specificity）は第 4 章第 3 節第 2 項 A）分類モデルの評価指標 を参照

表 2 -3 シミュレーションによる rf の結果の概要

sampling ratio		0.1	0.2	0.4	0.6	0.8	1
siz		64	128	255	383	510	638
accuracy	Min.	0.682	0.766	0.814	0.821	0.828	0.832
	1st Qu.	0.770	0.796	0.825	0.828	0.832	0.832
	Median	0.788	0.818	0.832	0.832	0.836	0.836
	Mean	0.781	0.811	0.830	0.833	0.836	0.834
	3rd Qu.	0.803	0.828	0.836	0.836	0.839	0.836
	Max.	0.828	0.832	0.847	0.843	0.847	0.836
	3rd Qu./1st Qu.	1.043	1.041	1.013	1.009	1.009	1.004
sensitivity	Min.	0	0.227	0.360	0.373	0.427	0.427
	1st Qu.	0.307	0.400	0.427	0.427	0.427	0.427
	Median	0.427	0.440	0.440	0.440	0.440	0.440
	Mean	0.411	0.431	0.436	0.448	0.452	0.439
	3rd Qu.	0.573	0.493	0.453	0.453	0.440	0.440
	Max.	0.720	0.547	0.480	0.613	0.653	0.520
	3rd Qu./1st Qu.	1.870	1.233	1.062	1.062	1.031	1.031
specificity	Min.	0.668	0.879	0.950	0.920	0.915	0.950
	1st Qu.	0.879	0.930	0.970	0.975	0.985	0.985
	Median	0.960	0.960	0.985	0.980	0.985	0.985
	Mean	0.920	0.954	0.979	0.978	0.982	0.984
	3rd Qu.	0.990	0.980	0.985	0.985	0.990	0.985
	Max.	1	1	1	0.995	0.995	0.985
	3rd Qu./1st Qu.	1.126	1.054	1.016	1.010	1.005	1.000

第 2 節 データの不足を補う（水増し）

データの不足を補う方法には大きく分けて似たデータを流用する方法と人工的にデータを作る方法がある．前者はデータが画像，音声，自然言語（文章）など非構造化データを得意とする深層学習の手法で用いられる．深層学習は，まず，データから特徴量の抽出を行う隠れ層を構築したのち，抽出された特徴量から推定や分類の学習が行われる．このため，特徴量の抽出では実際に推定や分類の対象となるデータ（「目的データ」）とは異なるデータ（例えば，目的データの眼底写真の代わりに花や動物の写真を用いる．仮にこれらを「代替えデータ」とする）を代用することが可能な場合があり，代替えデータとして利用可能なデータにより学習されたモデルがあれば，これを利用して少ない目的データのみで学習させることが可能である（第 5 章第 3 節転移学習，第 4 節ファインチューニング参照）．後者は，目的データにノイズを加えたり，部分的に置換するなど，様々に加工してデータを水増しする方法である（「水増しデータ」）．画像や自然言語は水増しすることが容易で，必要な目的データを大きく節約することが可能である．対して，rf やサポートベクターマシン（Support

Vector Machine：SVM）など，深層学習以外の機械学習手法では代替えデータを利用することはできない．

また，構造化データの場合も代替えデータを用いることはできないが，水増しは可能で，欠測値を補完して利用可能なデータを増やす，連続変数の説明変数にノイズを加える，クラス変数は出現頻度を重みとしてランダムに発生させるなどがある．しかし，このような単純な手法ではデータ間の相関関係は無視されるためデータを水増しする効果は低い．不均衡データについては，こういった問題を改善する手法としてマイノリティ合成によるオーバー・サンプリング（synthetic minority over-sampling：SMOTE）[19]があり，Python や R にもライブラリが提供されており簡単に利用できる[注 14]．

構造化データに限らず，画像データなど非構造化データにおいてもノイズを加えるなどによりデータを水増し・拡張する（Data Augmentation）手法が用いられる．データの種類と水増しの方法を簡単に整理すると，

- 構造化データ（テーブルデータ）は不均衡データであれば SMOTE（派生した多くの手法がある[20]）など k 近傍法や主成分分析などを用いて実在するデータに近いダミーのデータにより水増しする．
- 画像データは左右反転，傾ける，拡大縮小，フレームをずらす，ノイズ追加等を行う．
- テキストデータは異音同義語や類語などルールベースで単語置換を行う．
- 音声データは数値配列にしてノイズを乗せたり伸ばしたりする．

が行われる．

第 3 節 学習データの入手について

第 1 項 目的データ/代替えデータ

先に述べた通り，代替えデータによる学習が可能な場合とそうでない場合で対象が大きく異なるので分けて説明する．

我々製薬企業が機械学習を行う場合，対象となるのは化合物（医薬品の候補），ターゲットタンパク，医療情報（画像，波形，検査値，診療録など），活動（生活）情報，音声，表情，SNS 等である場合が多いであろう．いずれにしても過去の創薬や臨床開発の過程で多少は自社内に蓄積したデータがある場合は少なくないであろうが，多くの場合はデータ量，多様性が不足するため外部からの取得が必要となる．更に，データの質についても考慮する必要がある．目的データは実際に予測・分類するデータと粒度，バラツキなどデータの特性が同質である必要がある．例えば乳幼児の臨床上の予測を目的とする場合に成人の臨床データを用いることは適切ではない．また，複数の項目からなる複雑な構造化データを既存デ

注14 Python は imbalanced-learn，R は DMwR や smotefamily.

ータから得ようとする場合，多くの項目で欠測が生じていることが少なくない．また，時系列データの場合，測定されている時間間隔が同じである必要がある．このように新たにデータを取得した場合，データの質を確保するために人の手で加工する必要があることを考慮しなければならない．

以下に外部からデータを取得するに際し，ソースとなり得る候補を記載する．

A） オープンソース

機械学習分野では，MIT などのライセンスのオープンソースが多くあり，その中に目的データがあれば利用可能である．

学習データは目的に応じて様々である．一般的な機械学習データはラベル付きの動植物，食べ物，生活用品などの画像や SNS データなどに大規模なものが公開されている．製薬企業が必要とすることが多い医療系のデータについても，様々なデータが公開されている．例示として複数のデータを掲載または紹介（リンク）しているポータルサイトを表２-4 に紹介する．これらオープンデータには非商用利用であれば出典を明示することで提供者に許可を得たり，契約することなく無料で利用可能な Creative Commons（https://creativecommons.jp/ ：CC と略される），MIT License（https://opensource.org/licenses/mit-license.php）のものが多いが，利用時にユーザー登録が必要なものもある．学習により作成したモデル（AI）を販売するなど商用利用する場合はデータ提供者に確認が必要である．

表２-4 機械学習に利用可能なデータのポータルサイト

FDA Adverse Events Reporting System (FAERS) 米国 FDA（米国食品医薬品局）が運営している有害事象（副作用）の自発的報告システム．米国内のみでなく，米国で発売されている医薬品の米国外の報告も含まれている．報告者は医療専門家，患者，製薬企業など様々な関係者による膨大なレポートデータが集約されている．報告件数は約 1200 万件，薬剤は約 1 万 5 千種類ある． https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html
PubChem 米国 NIH 傘下の NCBI によって管理・運営が開始された化合物データベース．原子数 1000 以下かつ 1000 結合以下の比較的小さな分子が収録されている．Compound, Substance, BioAssay の 3 つのデータベースからなる．Python にはライブラリ（PubChemPy）から API を用いて検索と分子式，分子量，xlogP，tPSA，SMILES などのデータの取得ができる． https://pubchem.ncbi.nlm.nih.gov/
ChemSpider 英国化学会（RSC）が所有する化合物データベース．Python にはライブラリ（ChemSpiPy）から API を用いて検索とデータの取得ができるが，ウェブサイトにユーザー登録と API の利用申請が必要． http://www.chemspider.com/

SIDER 4.1 : Side Effect Resource 市販薬の添付文書に記載された副作用情報を収集したデータベース．バージョン 4.1 では 1430 件の薬剤が登録されている．データには，治験の副作用頻度のテーブルが含まれている． http://sideeffects.embl.de/
The PubChemQC Project 理化学研究情報基盤センターが管理している国産のデータベース．PubChemQC は，PubChem に登録されている分子をターゲットにしており，約 320 万個の分子について，半経験的分子軌道法(PM3)，HF/STO-6G，B3LYP/6-31Gd，更に TD-DFT/6-31+Gd で励起状態計算を行った構造最適化計算の結果が提供されている． http://pccdb.org/
DrugBank カナダのアルバータ大学で管理・運営されている，医薬品及び医薬候補化合物とそれらターゲットに関するデータベース．約 13,000 件の化合物について，薬理的，薬事的データ，ターゲット（配列，構造，パスウエイ）の詳細情報を収集，統合したバイオインフォマティクス及びケモインフォマティクスのリソース．薬剤エントリー約 4,800 件のうち，FDA 承認小分子薬剤が 1,300 件以上が含まれている．各エントリーにつき 100 以上のデータ項目が存在する．データのダウンロードには会員登録が必要． http://www.drugbank.ca/
FMODB: The database of quantum mechanical data based on the FMO method FMO Drug Design（FMODD）コンソーシアムによるフラグメント分子軌道法（FMO 法）を用いた種々のたんぱく質の量子化学（第一原理）計算結果のデータベース．CC BY-SA 4.0 license で利用可能． https://drugdesign.riken.jp/FMODB/
ImageNet 恐らくもっとも有名な機械学習用画像データベース．1 万以上のシンセット（シノニム）からなる WordNet [注 15] 階層の各階層に対応して編成されている． 1 階層に平均約 500 枚の画像が提供されている． http://www.image-net.org/
Visual Genome 画像数は多くはないが，画像内の対象物の矩形領域それぞれに短いキャプションがアノテートされ，相互の関連性や生じてることなど，画像 1 枚 1 枚に多くの情報が付与されている． http://visualgenome.org/
Places 空港，寝室など，合計で 400 以上のユニークなシーンカテゴリ（場所や状況）で構成される 1,000 万枚以上の画像が含まれている．このデータセットは，シーンごとに 5000～30,000 のトレーニング画像を備えており，畳み込みニューラルネットワーク（CNN）を使用して，さまざまなシーン認識タスクのディープシーンフィーチャを学習できる． http://places2.csail.mit.edu/index.html
【18 個掲載】機械学習に使える生命科学・医療関連のデータセットまとめ 一般的なデータセット，画像データセット，ゲノムデータセット，病院のデータセット，癌に関するデータセットがリストされている． https://lionbridge.ai/ja/datasets/18-free-life-sciences-medical-datasets-for-machine-learning/

注15 プリンストン大学が始めた単語を語彙概念でグループ化，関連付けしたデータベース（https://wordnet.princeton.edu/）．独立行政法人情報通信研究機構によって日本語版が作成，公開されている（http://compling.hss.ntu.edu.sg/wnja/）

arXivTimes / datasets 機械学習を行う際に利用可能なデータセットについてまとめている． https://github.com/arXivTimes/arXivTimes/tree/master/datasets
Microsoft Azure AI Gallery Microsoft Azure ML の学習済モデルのライブラリ，カテゴリに Healthcare があり，心疾患の予測や乳がんの分類等の学習例が公開されている．使用するデータは外部の OPENDATA または Microsoft Azure ML が公開しているデータ． https://gallery.azure.ai/browse?industries=["5"]
UCI Machine Learning リポジトリ カリフォルニア州アーバイン:カリフォルニア大学，情報・コンピューター サイエンス学部 ．検索にはコツが必要. https://www.trifields.jp/uci-machine-learning-repository-datasets-956 に日本語で紹介されている． https://archive.ics.uci.edu/ml
BROAD Institute Cancer Program Datasets 脳腫瘍，白血病，黒色腫など 98 の研究プログラムのデータセット．ゲノム（SNP，DNA，RNA）のデータが多数ある． http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi
Kaggle Datasets Kaggle のコンペティションで使用されたデータセットを検索できる． https://www.kaggle.com/datasets
Harvard Dataverse ハーバード大学が公開している研究データ．Medicine, Health and Life Sciences のカテゴリには 2000 件以上のデータセットがある． https://dataverse.harvard.edu/
Microsoft Research Open Data Microsoft 社が公開する無料のデータセットのページ．BETA 版．医学系のデータはない． https://msropendata.com/
Medical Data for Machine Learning 様々なデータセットのリンクリスト．UCI Datasets も肝臓，乳がん，心疾患など分類されていてわかりやすい． https://github.com/beamandrew/medical-data
List of Medical Datasets 主に医療画像データセットのリンクリスト．ICU の患者データなど少数だが画像以外もある． https://github.com/adalca/medical-datasets
the National Biomedical Image Archive Project 無料のオープンソースサービス及びソフトウェアアプリケーション．ユーザーが診断医療画像を安全に保存，検索，及びダウンロードできる．がん画像，臨床データ及びゲノムデータを統合した検索可能な全国リポジトリが提供されている． https://github.com/NCIP/national-biomedical-image-archive
Google Cloud 一般公開データセット USA Disease Surveillance，Medicare Data などがある．患者ごとの臨床データはない． https://cloud.google.com/public-datasets/?hl=ja
Boos-Stefanski Variable Selection Home scikit-learn の糖尿病サンプルデータなどが掲載されている． https://www4.stat.ncsu.edu/~boos/var.select/

B） 販売されているデータソース

化合物，ターゲットタンパクなど，国・地域に無関係なデータは国内外を問わず販売されているデータが利用可能である．対して医療情報等はその種類，目的により国・地域が限定される場合がある．現在，日本国内で機械学習に利用可能な医療情報の提供は限られている[注16]．医療情報は大きく分けてレセプト（医科・歯科・調剤の診療報酬明細書），DPC データ，医療画像，臨床検査（検体検査）データ，診療録（病歴，所見，経過などのフリーテキスト），看護記録，退院サマリなどがある．このうち，レセプト，DPC データ及び規模は限られるが臨床検査データについては商用のソースがある．また，商用ではないが，臨床検査データを含む大規模なデータソースとして，独立行政法人医薬品医療機器総合機構が運営している MID-NET がある．MID-NET はレセプト＋DPC＋臨床検査データからなるデータベースであるが，民間企業による利用は RMP に基づく製造販売後データベース調査に限定され，利用もオンサイトセンターにおける MID-NET システム内での集計に限定されているため機械学習のソースとして利用できない．対して，やはり商用ではないが，独立行政法人国立病院機構が構築運営している NCDA（国立病院機構 診療情報集積基盤 https://nho.hosp.go.jp/cnt1-1_000070.html）は MID-NET 同様にオンサイトセンター内での集計に限定されるが，利用者が必要なハード・ソフトを持ち込むことが可能であり，機械学習のソースとして使用することが可能である．また，次世代医療基盤法に基づく認定匿名加工医療情報作成事業者の第 1 号が 2019 年 12 月に認定されており，今後利用可能なソースの拡大が期待される．

C） 自社以外の企業が保有するソース

多くの製薬企業はこれまで創薬過程で評価した化合物，ターゲットに関する多くのデータ，多くの臨床（治験）データが蓄積されているが，目的とする疾患分野が異なるなどの理由により自社では利用価値が低いが，同じ分野で開発を行う他社では利用価値が高い場合がある．近年，これらについてデータシェアリング（Clinical Trial Data Sharing[21]）の動きがある．産学連携全国がんゲノムスクリーニング事業（SCRUM-Japan）が良い例であろう．製薬企業 17 社，全国の 250 以上の医療機関が参加し， SCRUM-Japan に臨床・ゲノム情報を登録している．登録されたデータは，セキュアな環境で医療機関・製薬企業との間で共有・二次利用を行い，がん新薬の創出や様々ながん治療の研究に広く利用されている．また，組織外の知識や技術を取り込むため，他企業や他業種と連携することにより，付加価値の高い成果物が期待できる,オープンイノベーションの動きも近年活発である．しかし，双方の認識のずれやプロジェクト管理の甘さが原因で訴訟に至るケースもある．うまく機能させるためには，企業文化の違いや専門分野の違いを認識した上でのマネジメントとコ

注16 機械学習に用いられるという区分はないが，日本薬剤疫学会「日本における臨床疫学・薬剤疫学に応用可能なデータベース調査」にデータベースの規模，内容などの概要がまとめられている．https://sites.google.com/view/jspe-database-ja/

ミュニケーションが重要である[22]．そのほか，化合物や非臨床のデータであれば知財，工業権の整理が，ヒトに係るデータであれば個人情報保護法・倫理面での整理が必要となる．

第 4 節 データ確認（視覚化，探索）

機械学習で扱うデータは大きく分けて構造化データと非構造化データに分けられる．データの確認という視点では，非構造化データは画像（静止・動画），音声，文章に，構造化データは連続変数（尺度），カテゴリカル（クラス）変数，時系列データに分けられる．ここでは代表的なものとして画像データ，自然言語と時系列データについて説明する．

第 1 項 画像データ

画像の場合はフォーマット（TIF，GIF，JPEG など），サイズ（縦横のピクセル数），色（白黒，グレイスケール，カラー），対象物（人物，花，動物など），加えて動画はフレーム数といった「規格」を確認する．特に規格が統一されているか，混在するかは重要である．次に対象物にピントが合っているか，フレーム（全体が写っているか，切れていないか，中心にあるか），多様性（人の顔であれば性別，年齢，人種，顔の向き，全く関係ない画像も含むなど）などを確認する．前半の規格は画像ファイルのプロパティ（Exif データなど）を検索することで一括して確認，フィルタリングが可能である．後半の品質関連は簡単に検索はできないが，一般に学習データは数千から数万となることが少なくないため人の目で確認することも容易ではない．このため，例えば，WEB 上の画像データを Google や Bing の Image Search API（Application Programming Interface）を用いて収集して学習済の画像認識モデルである OpenCV（Python のライブラリがある）を用いた画像認識により対象となる画像の選択と対象箇所（例えば人の顔など）の抽出（crop）を行い，サイズ・解像度を統一，JPEG に変換して保存するという一連の作業を Python によるスクリプトを作成して行うことで，ある程度絞り込むことができる．他の方法を含めて収集・入手した画像データからランダムに抽出した数十から数百の画像を人の眼で確認する．学習の邪魔になる画像が多く含まれている場合は，次の前処理を含めて最終的には人の眼で確認することになるため，可能な限りあらかじめ絞り込んでおくのが望ましい．例えば，確認した画像から数十枚の望ましい画像と除きたい画像を選び，画像を 1000 種類に分類できる VGG16 等の汎用的な画像分類モデルの中間層から得られる特徴量を用いて入手した画像から類似した画像を検索することで学習に好ましい画像と好ましくない画像に分けることができる．

第 2 項 自然言語（文章）

主要な SNS には機械学習を利用した検索 API（クローラー）が利用可能であるので，学習目的の文章に特異的に含まれている単語（病名や医薬品名など）や一般的であるが含まれることが多い用語（つらい，痛いなど）を検索条件とすることである程度絞り込むことができる．このように検索して収集したデータや他の方法で入手したデータは，画像同様に最終

的には人が読んで学習に適切なデータであるか確認する必要がある．しかし，すべてを読むことは不可能なため，通常はランダムに抽出した数十の文章を読んで確認するほか，文章の長さ（文字数），テキストマイニングツールを用いて単語の出現頻度などの統計量やワードクラウドや共起ネットワークなどによる視覚化により確認する．

第 3 項 構造化データ（時系列データ）

構造化データは列と行からなるデータで，多くは列をカンマ，タブ，スペースなどで区切り，行は改行記号で区切られたデータである．機械学習の分野ではデータのサイズが大きく，文字コードに UFT-8，改行コードに LF（Windows 系の標準改行コードは CR+LF である）が用いられていることが多い．このため，メモ帳や MS Excel などでは開くことができない，開けた場合でも，文字コードなどを手動で変更しなければ，正しく表示できない場合がある．また，時系列データの場合，日時とデータ項目を表すコードと値の繰り返しとなっている場合もある．加えて，機械学習に用いる構造化データは製造現場，日常生活，通信，インターネットの利用など現実世界で生じるいわゆる RWD である場合が多い．特に医療・健康系の RWD ではデータの欠損（欠測），単位，フォーマットの不統一などがあるので，データの確認は重要である．

例として，第 4 章で扱うデータを用いて説明する．データは患者ごとにファイルが分けられており，表 2 -5 のように日付，時間，項目コード，値のタブ区切りとなっている．70 ファイルあるので，扱いやすくするため，患者 ID の列を追加して 1 ファイルに結合させる．そして患者ごとに期間，日時の繰り返し数，項目ごとのデータ数，項目ごとのデータの範囲，患者ごとの各項目のデータ数，各項目の要約統計量，患者ごとの値の推移（グラフ）などを確認する．詳細は第 4 章第 2 節第 1 項で紹介している．

表 2 -5 糖尿病患者データの例

04-21-1991	9:09	58	100
04-21-1991	9:09	33	009
04-21-1991	9:09	34	013
04-21-1991	17:08	62	119
04-21-1991	17:08	33	007
04-21-1991	22:51	48	123
04-22-1991	7:35	58	216

第 5 節 前処理（データの加工）

構造化データであれば，データの構造を扱いやすいように変形する，値の単位や書式（例えば日付など）を揃える，項目を組み合わせて（連結，比や差など）新たに値を作成する，時系列データでは日時を相対的な値にする，前値との差，傾きやトレンドを計算するなど，回帰分析手法による統計解析でも通常用いる操作が行われる．画像データではフォーマッ

ト（JPEG など），解像度や色調を統一するなどが行われる．前処理は用いる機械学習の方法によって必要な事項が異なり得るため，どのように機械学習をさせるのかを考えてから行う必要がある．以下では構造化データを中心に説明する．

第 1 項 データ構造の変形

構造化データの場合，複数のファイルに分かれているデータを 1 つにまとめたり，複数行のデータを 1 行にするなど，学習の手法で用いやすい形式にデータ構造を変形させる．第 4 章の糖尿病データは既に説明した通り，患者ごとにファイルが分かれており，血糖値，インスリン投与量，低血糖イベントなどデータ項目が各々1 行である．学習は 1 回のインスリン投与と空腹時（食前）血糖値を起点として次のインスリン投与，空腹時（食前）血糖値の直前までを 1 組としてその間の低血糖の発生の予測を行う．このため，患者の ID の項目を追加して 1 ファイルに併合して，1 組のデータを 1 行に加工する．詳細は第 4 章を参照いただきたい．

第 2 項 値の変換，標準化

単位を揃える，連続変数を標準化・正規化する，対数・逆数等変換する，生年月日から年齢を計算する，身長と体重から BMI を計算する，異常値を削除するなどを行う．機械学習で用いる手法によっては，連続変数の項目（特徴量）の範囲が揃っている，分布が正規分布や一様分布である方が効率よく学習できる場合がある．そのため，標準化・正規化が行われる．

- 標準化：特徴量の平均を 0，分散を 1 とする変換である．連続変数が X，X の平均を μ_x，標準偏差を s_x により表すと，標準化後の$\mathbf{Xstd} = (\boldsymbol{X} - \boldsymbol{\mu_x})/\boldsymbol{s_x}$ により得られる．
- 正規化：値を一定の範囲に収める変換である．通常，0～1 または-1～1 に変換される場合が多い．例えば 0～1 の範囲に変換するには，X の最小値を X_{min}，最大値を X_{max} ，X の個々の値を X_i と表すと，正規化後の $\boldsymbol{X_{norm \cdot i}} = (\boldsymbol{X_i} - \boldsymbol{X_{min}})/(\boldsymbol{X_{max}} - \boldsymbol{X_{min}})$ により得られる．
- ロバスト標準化（robust z score）：標準化及び正規化は元の変数の分布形状が維持されるため，外れ値の影響は緩和されない．外れ値に強い（頑健な）手法として四分位の 75%点（X_{3q}）から 25%点（X_{1q}）を引いた四分位範囲（interquartile range, IQR）を用いるロバスト標準化が用いられることがある．計算は，IQR÷正規分布の四分位範囲（1.3489）により得られる正規四分位範囲 normalized IQR,
  $\boldsymbol{NIQR} = \left(\boldsymbol{X_{3q}} - \boldsymbol{X_{1q}}\right)/\mathbf{1.3489}$を用い，中央値（median）を X_m で表すと，
  $\mathbf{z} = (\boldsymbol{X_i} - \boldsymbol{X_m})/\boldsymbol{NIQR}$により得られる．
- 離散化：連続点数を任意の区間でグループ化する方法である．例えば 0 から 99 の連続変数で整数化したり，1 の桁を四捨五入する，5%点ごとに区切る，時間を時間帯で

朝，昼，夜，月を季節に変換するなどは離散化の手法である．機械学習では“ビンに区切る”，“ビニング”と呼ぶことがある．1 つの変数として変換することも，第 3 項のダミー変数とすることもある．

標準化，正規化は異常値の処理，単位の統一，対数・逆数等変換などを行った後に行う必要がある．事例の糖尿病データの一部（第 4 章の学習データ）の空腹時血糖値を例として紹介する．血糖値は通常 50mg/dL から 400mg/dL ほどの範囲の値であるが，正常値は 100mg/dL の前後の狭い範囲で右側に広い分布である．このことから，データに 100mg/dL を中心とした意味を持たせるという発想により，血糖値を 100 で割り，対数変換する（つまり 100mg/dL は「0」となり，低い値は負となる（図 2 -4）．Log_{10}(血糖値÷100)）変換した場合と生の値をそのまま用いた場合のロジスティックモデルによる低血糖イベントの判別結果は ROC 曲線の AUC は共に 0.67752 と全く同じであるが，混同行列は表 2 -6 のとおりで変換により再現率は大きく改善し，代わりに特異度は大きく低下している．わかりやすく身近なロジスティックモデルを用いて紹介したが，機械学習の手法には線形モデルを用いることが少なくないため，このような変換により結果が変わる場合が少なくない．なお，決定木に代表される，閾値による区分けを行う非線形手法では値の線形変換は結果に影響しない．先ほどの血糖値データに対して，最も単純な決定木の手法である CART を用いると，生のデータ，対数変換したデータのいずれも ROC 曲線の AUC は 0.6702 であるとともに混同行列も表 2 -7 のように全く同じになる．

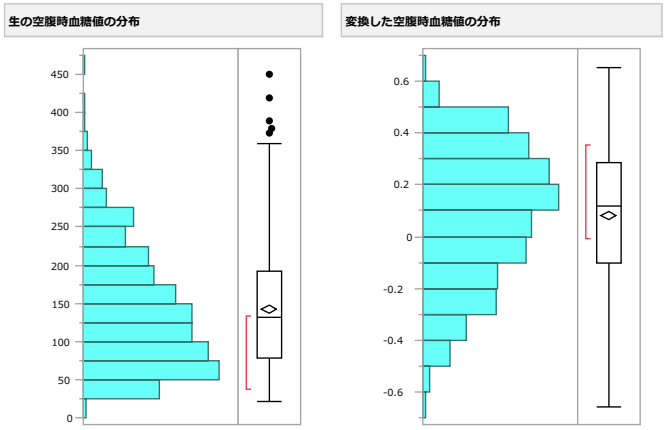


図 2 -4 血糖値の生のデータと変換したデータの分布

表 2 -6 ロジスティック回帰分析の結果の混同行列

		変換なし			変換あり		
		あり	なし	再現率/特異度	あり	なし	再現率/特異度
実測	あり	2	293	0.0068 0.9986	74	221	0.2508 0.9436
	なし	1	691		39	653	

表 2 -7 CART の結果の混同行列

		変換なし			変換あり		
		あり	なし	再現率/特異度	あり	なし	再現率/特異度
実測	あり	69	226	0.2339	69	226	0.2339
	なし	24	668	0.9653	24	668	0.9653

第 3 項 ダミー変数の作成

順序性がないマルチプル（要素が 3 個以上）なカテゴリカル変数を要素ごとに分解して二値（有無，1 と 0）の変数とする．例えばデータに居住地の都道府県の列がある場合，各都道府県の 47 の列を作成し，それぞれ該当する場合に 1，該当しない場合に 0 とする加工を行う．このように要素数と同じ数のカテゴリカル変数で表す場合をワンホットコーディング（One-Hot encoding）と呼ぶ．また，任意の 1 つの要素（例えば北海道）を全ての変数が 0 の場合とすることで，要素数-1 のカテゴリカル変数で表すことができる．この場合をダミーコーディング（dummy coding）とよぶ．多くの機械学習のソフト，ライブラリは内部で自動処理できる場合が多いので，前処理としては，必ずしも必要ではない．順序性のあるカテゴリカル変数には順序を与える（整数に変換する）．多くの機械学習のソフト，ライブラリはカテゴリカル変数を factor（data.frame のデータ型）として扱い，内部的に順序変数（連続した整数）に変換する機能がある．連続変数であっても，結果変数との関係が非線形であったり，特性が段階的に変化する場合などはカテゴリ化（例えば年齢を乳児，小児，成人，高齢者とするなど）を行い，ダミー変数に変換する場合がある．

第 4 項 分散がゼロ（に等しい）予測変数

全てあるいは多くのデータが同じ値を取る変数がある場合，学習の過程でサンプリングされる際（交差検証，ブートストラップなど）に全てのデータが同じデータになり，計算不能となる（クラッシュする）場合があるため，削除する又は学習過程のサンプリングに工夫が必要な場合がある．第 4 章の例では，48：Unspecified blood glucose measurement，57：Unspecified blood glucose measurement 59：Post-breakfast blood glucose measurement，61：Post-lunch blood glucose measurement，63：Post-supper blood glucose measurement，66：Typical meal ingestion，67：More-than-usual meal ingestion，68：Less-than-usual meal ingestion，69：Typical exercise activity，70：More-than-usual exercise activity，71：Less-than-usual exercise activity，72：Unspecified special event をデータが少ない，データがない患者が多いため削除している．

第５項 互いに相関する予測変数

身長と体重など互いに相関関係の大きい変数を共に用いると多重共線性により不安定となる場合があるため一方を除く，あるいは身長と体重であれば肥満度（BMI）といった別の意味のある変数に変換するなどを行う．第 4 章の例では，長時間型の 2 種類のインスリン（NPH insulin，Ultra Lente insulin）の使用量を合計した UltraLente_NPH_insulin_dose を作成している．

第６項 欠測値の処理

多くの機械学習の手法では，1 つでも特徴量に欠測値のある場合，レコード全体が学習に使用されない．このためあらかじめ除外するか，代入（補完）を行う必要がある．カテゴリカル変数であれば欠測値のカテゴリ（NaN など）を作成することもできる．代入は前処理として行う（つまり予めデータセットに入力する）場合と，学習中のサンプリングごとに確率的に代入する方法がある．いずれの場合でも，平均値・中央値，回帰，K 近傍法，ランダムフォレストなどが用いられることが多い．

第７項 粒度の変更（グループ化）

主に順序のないカテゴリカル変数で多くの要素がある場合にグループ化して要素の数を減らすことで学習の効率，汎化性能が良くなることが期待できる．例えば自由記述的な病名を ICD-10 コードによりコーディングする，更に ICD-10 分類階層を上げる，少数（数個）の要素が大半を占める場合，それ以外の要素をまとめるなどが行われる．この処理はダミー変数を減らすことであり，次の変数の削減にあたる．連続変数の場合でも，日付を年月に丸める，年齢を 5 歳刻に丸めるなど，丸めの処理などによる粒度の変更が行われる．

第８項 変数の削減

深層学習では特徴量の抽出により情報の要約が行われるため前処理として人手による変数の削減が不要である場合が多い．一般的な機械学習の手法においては，予測変数が連続変数である場合，多変数から主成分分析，独立成分分析などにより情報を縮約し変数を減らすことで学習効率が良くなることが期待できる．

第９項 テキストデータの前処理

非構造化データであるテキストデータ（文章）では，まず単語に切り分ける必要がある．欧米の言語の多くは単語をスペースで区切るため，特に作業を要しないが，日本語は単語の区切りがないため，大きな作業となる．前処理の全体は図２-5 の様になる．クリーニングでは，その対象のノイズはデータソースや学習の目的によって変わるが，ブログや SNS などをソースとする場合，html のタグ，url など，文章に直接関係ないデータを bs4

（BeautifulSoup） や lxml[注17]などのライブラリを用いて消去する．単語への切り分けは MeCab，ChaSen，Juman，Janome[注18]など「形態素解析器」と呼ばれるライブラリを用いて行うことで同時に品詞情報の付加，単語の正規化で行う活用形を原型に変換することができる（第５章第２節第２項参照）．この際，複合語による（固有）名詞，新語，専門用語などがバラバラにされてしまう傾向があるが，対応した辞書を用いることで軽減できる．

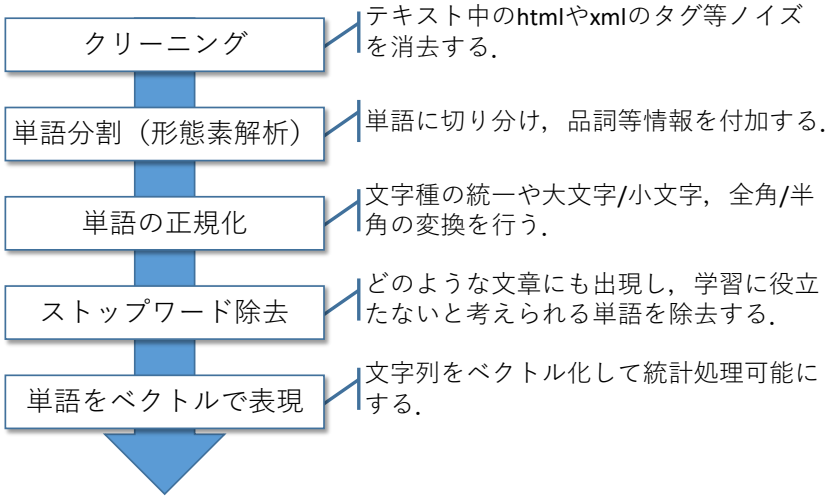


図２-5 テキストデータの前処理の概要

単語の正規化は活用形を原型に変換するほか，数字をアラビア数字に統一する，半角カナを全角に変換する，アルファベットを半角小文字に，日付の書式を yyyy-mm-dd 形式に統一するなどの他，辞書を用いてカタカナ英語や漢字，カタカナ，ひらがな，ローマ字，略称，ロゴなど多様な表記が用いられるブランド名や国名の統一などを行う．そして接続詞，助詞など，どのような文書にも出現して学習に役立たない単語（ストップワード）を形態素解析で付加された品詞情報を用いたり，出現頻度などを用いて識別，除去する．単語のベクトル表現（変換）は，Word2Vec[注19][23]等のライブラリを用いて文章をベクトル化する工程で，単純な方法としては出現単語数の次元数のベクトルを用い，各単語を特定の次元に割り当てて 1,0 で表現するワンホット（One-hot）表現がある．この方法は単語数だけベクトルの次元数が必要なうえ，異なる単語間ではベクトルの内積が全て「0」となってしまうため，演算に向いていない．これに対して，一般には各次元が 0～1 の実数からなる数十から数百

注17 bs4 及び lxml は WEB から HTML データを取得，解析するライブラリ，解析には HTML パーサーを使用するが，Python 標準の html.parser より lxml は効率が良く，両者を組み合わせて利用される．

注18 Mecab は最も有名な日本語の自然言語処理器（形態素解析エンジン）．平均的に ChaSen, Juman より高速に動作し，辞書やコーパスの種類に依存せずに利用可能な汎用的な設計となっている（http://taku910.github.io/mecab/）．Juman は WEB テキストから辞書情報を自動獲得でき，Wikipedia から抽出した辞書も利用できる，UTF-8 対応，辞書情報はあらかじめ内包されているため，機械学習に便利である．Janome は Pure Python でつくられており，MeCab などの外部エンジンは不要，辞書を内包しているので容易に利用できる．

注19 Google に在籍していた研究者であるトマス・ミコロフ氏らにより開発された，２層のニューラルネットワークを用いて単語を低次元のベクトルを用いて分散表現を行うライブラリ．

次元のベクトルを用いて単語の意味を扱う分散表現が用いられる．分散表現では，特定の 1 次元のみが「0」以外の値を取るのではなく，全ての次元が 0 以外の値を取り得る[注 20]．分散表現の方法には，Word2Vec で用いられている同じ文脈に出現する単語は似た意味であるとする分布仮説，WordNet [注 15] で用いられている意味辞書を用いて，関連している単語群は似たベクトルになるようにファインチューニング（fine-tuning）するレトロフィッティング（retrofitting）[24]等がある．分布仮説では辞書を必要としないが，対義語や多義語の扱いに弱い．このような単語のベクトル変換は自然言語処理の基本的な手法であるが，類似文章の検索など目的によっては単語のベクトル変換を行わず，文章中に出現する各単語をベクトルの次元として単語ごとの出現頻度（回数）を次元の値とするバグオブワード（Bag of Words：BoW）と呼ぶ手法なども用いられる．

第１０項　　非構造化データのアノテーション

画像や文章などの非構造化データに対して，機械学習を行いやすいように情報を付加（意味付け，印付け）する作業である．これには，データへの正解のラベル付けによる教師データの作成，画像のマーキング，あるいは特徴量の付与など様々な方法が含まれる．代表的なアノテーションの例としては以下のものがある．

- 画像中の対象物を枠で囲む，人の顔であれば目鼻口角などをマーク（座標データ）する．（画像・映像アノテーション）
- 画像に写っている対象物の種類，数などのラベル（構造化データ）を付ける．（エンティティアノテーション）
- 花や動物の画像に対して種類名のラベルを付ける（セマンティックアノテーション）
- 文章全体の意味分類，ポジティブ/ネガティブ/ニュートラルなどのラベルを付ける（インテント抽出）
- 文書中の品詞に品詞名のタグをつける（文節チャンキング）
- 文章中の人の名前，地名，商品名などの固有名詞，感情を表す単語などをそれぞれ区別したタグで挟む（タグ付け）．（固有表現認識，エンティティアノテーション）
- 文書中の単語間に関連付けを行う（エンティティリンキング）

注20 分散表現について，第 5 章第 2 節第 1 項 B）で説明している．

第３章　学習のためのデータ分割と学習方法

深層学習を含め，機械学習はモデルを作成（パラメータを探索）するデータと，その結果を確認するデータに分けて行われる．一般に前者を学習（用）データ，後者をテスト（用）データと呼ぶ．学習では探索的に同じデータを繰り返し用いて最適なモデルを作成するため，学習データの特徴に最適化されすぎ，他のデータでは性能が低下する過学習という現象が生じる，過学習を防止するため，学習に用いていないテストデータによりモデルを評価する．これにより，学習に用いていないデータでも性能が得られる「汎化性能」を適切に評価することができる．学習中に，モデルを評価するため，学習データを更に学習データと検証データに分割する場合がある．この場合，同じ“学習データ”という用語が用いられることもあるが，後者の場合の学習データと検証データを合わせて“開発データ”と呼ぶことがある．

データの分割及びパラメータを探索して学習する方法は複数あるが，以下では基本的手法であるホールドアウト検証法におけるデータ分割法の紹介ののち，代表的な手法を紹介する．

第１節 ホールドアウト検証法（Hold-out validation）

機械学習のデータは，モデルを作る学習データと学習した結果を最終的に確認するテストデータに分けて使用される．この分割方法には種々のものがあるが，ホールドアウト法は学習の前にあらかじめデータを分割して，学習中にテストデータを全く参照することがないシンプルな方法である．

データの分割はランダムに分けられることが多いが，単純に完全ランダムサンプリングを行うと，“偶然に”重要な変数の分布に偏りが生じ，学習及び検証に支障を来す場合がある．特に結果（目的）変数がカテゴリカル変数で低頻度の要素を含む場合は影響が大きい．実際の教師あり学習では，学習中にもモデルを評価するためのデータを取り分ける手法（後述）が併用される．この場合のデータは，モデルを作る学習データ，学習中にモデルを評価する検証データ，学習した結果を最終的に確認するテストデータに分けて使用される．検証データは交差検証法など，学習中に自動的に分けられるが，学習データ（検証データ含む．以下，合わせて「開発データ」と呼ぶ）とテストデータは人の手で分割することが少なくない．

第１項 結果に基づく層化分割

識別などの教師データ（結果変数）の分布が均等になるように分割する．事例の糖尿病データは結果変数である低血糖イベントは 4%程度しかないため，低血糖イベントの有無で層化してランダムに抽出する．

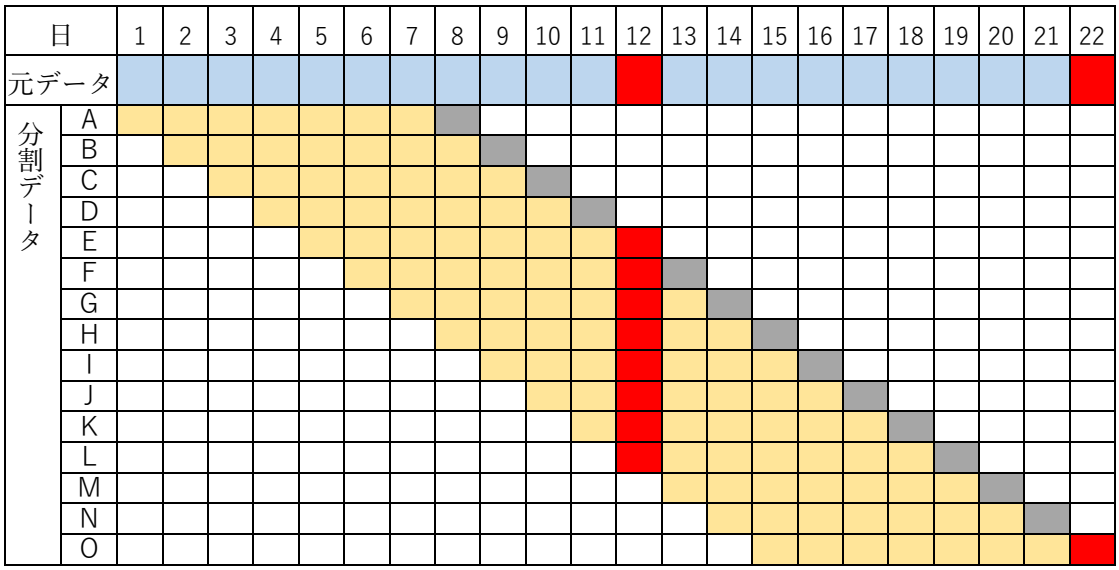
第 2 項 重要な意味のあるグループによるデータ分割

臨床試験の施設や，国・地域など，重要な意味のある層がある場合，層を考慮した（層化）分割を行う．また，機械学習の分類木に分類される手法では，学習データにない分類要素は扱えない場合があるため，開発データとテストデータの要素が揃うように分割する．第 4 章の糖尿病データは患者層別モデルを用いるため，必ずすべての患者が含まれている必要がある．このため，患者ごとにランダム抽出する．また，目的変数である低血糖イベントも少ないため，患者ごとに開発データ，テストデータ共にイベントありのデータが含まれるよう分割の層の条件としている．

第 3 項 時系列のデータ分割

事例の糖尿病データであれば，1 週間のデータから翌日の各投与直前の血糖値（空腹時血糖値）を予測する学習を行う場合，学習データは 8 日の連続データ（8 日目は目的変数）である必要がある．また，検証データ及びテストデータの目的変数に当たる日付のデータが学習データに含まれないようにする必要がある（検証データが学習データ，目的変数が予測変数に含まれることを「データリーク」などという）．表は，元データが 22 日分あり，12 日目と 22 日目に低血糖イベントが生じている．ここから学習，テストデータを分割すると，A～O の 15 組のデータが得られる（グレーが予測対象）．A～J の 10 組を開発用，K～O の 5 組をテスト用とすれば，開発データにテストの予測対象のデータを含めずに分割することができる．

表 3-1 時系列のデータのイメージ



第４項 予測子に基づいた分割

これは少し難しい概念である．履歴（観測）データのように似通った多量のデータが得られない場合，例えば分子構造のシミュレーションなどで用いられる．分割されたデータが同様に多様であるようにする手法で，非類似度最大化アプローチ（maximum dissimilarity approach）と呼ばれる．考え方としては全体のクラスタリングを行い，初めにランダムに１つのサンプルを抽出し，次にそのサンプルから最も離れたサンプルの抽出を繰り返してゆく[25]．

第２節 そのほかの検証法

第１項 交差検証法（cross validation）

ホールドアウト検証法では，学習中に学習に含まれないデータを用いた評価を行えず，学習の結果に過学習が生じていないかを確認できるだけである．また，評価は１回だけであるため，学習とテストに生じるデータの偏りにより評価結果の信頼度が低い．これに対して学習中に学習に用いるデータと評価するデータに複数回分割して評価を繰り返すことが行われており，交差検証法はその代表的な手法である．通常，分割する数をＫで表し，Ｋ分割交差検証法（k-fold cross validation）とも表記される．

Ｋ分割交差検証法では，図３-1 の青枠内のように，開発データをＫ個のブロック（サブセットとも呼ばれる）に分割[注21]し，そのうち１つのブロックを検証データとして除いて残りの K-1 個のブロックを学習データとして用い，検証データのブロックを学習データブロックと入れかえながら学習を行い，各回のパラメータの平均を評価する．ブロックの数は５や 10 がよく用いられる．ブロックの数を増やせば，学習の繰り返し数が増えるため成績が良くなることが期待できるが，学習量が多くなる（時間がかかる），検証データが少なくなるため，バランスを考えて設定する．交差検証法は単独でも用いられるが， 図３-1 のようにホールドアウト法と組み合わせてうことができる．

注21 Ｋ個のブロック全体＝開発データを Fold，テストデータを Hold-out と呼ぶ場合がある．

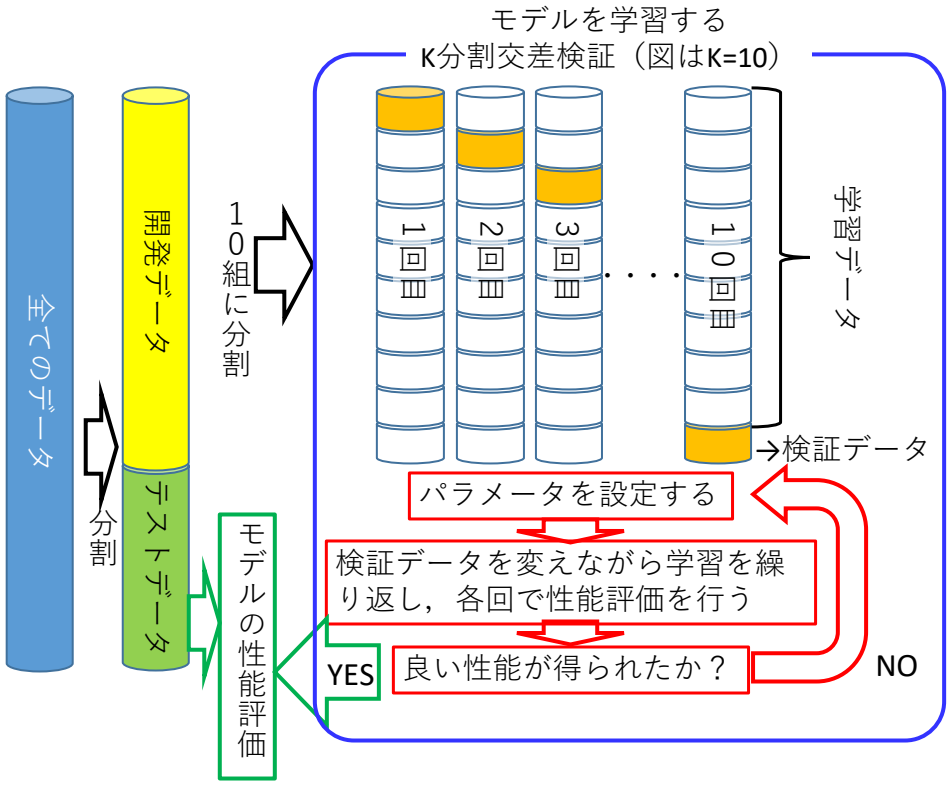


図３-1 データの分割と学習のイメージ

第２項 ブートストラップ法

交差検証法同様のパラメータチューニングでよく用いられる方法である．交差検証法ではあらかじめ，決めた数にデータを分割するが，ブートストラップ法（Bootstrapping）では，学習の都度，ランダムにサンプリングを行う．具体的には，開発用の n 個のデータから，ランダム復元抽出により n 個のデータをサンプリングする．復元抽出であるため，抽出された n 個には重複するデータが存在し，抽出されないデータが平均的に約 37%生じる[22]．抽出された n 個のデータで学習し，抽出されなかったデータで検証を行う．

第３項 ミニバッチ学習及びバッチ学習

ミニバッチ学習は深層学習でよく用いられる方法である．交差検証同様に学習データを同じサイズ（データ数）の複数のサブセットに分けて，順に１つのサブセットを用いて学習を行う．この時，サブセットに含まれるデータ数をバッチサイズと呼ぶ．例えば，2,000 件のデータを 10 個のサブセットに均等に分割すれば，バッチサイズは 2,000÷10=200 となる．そして，この場合，10 回学習を繰り返すと，2,000 件のすべてのデータが 1 回学習に

[22] 個々のデータが抽出されない確率は，$(1-1/n)^n$ である．

用いられたことになり，これを 1 エポック（epoch）と呼ぶ．対して，バッチサイズが 2,000 件，つまり 1 回の学習にすべてのデータを用いる場合をバッチ学習と呼ぶ．初めに「同じサイズ（データ数）の複数のサブセットに分けて」と説明したが，固定して分けるのではなく，ランダムな抽出（復元抽出）を繰り返す方法が用いられる．この場合も（すべてのデータが 1 回学習に用いられた保証はないが）バッチサイズが 200 件なら，10 回の学習で 1 エポックとなる．

第 4 項 オンライン学習

ミニバッチ学習でバッチサイズが最大となる学習用の全データとした場合はバッチ学習であった．反対に最小となるバッチサイズ＝1 とした場合は「オンライン学習」と呼ばれる．ミニバッチ学習はあらかじめ学習に用いるデータを固定する必要があるのに対して，オンライン学習は何らかのシステムなどから発生するデータを順次用いて継続的な学習が可能であるという特徴がある．

第 4 章　臨床試験データや RWD に機械学習を活用する実装例

この章では，構造化データを用いた二値分類の問題について，実際にデータを確認し，試行錯誤しながら特徴量の作成，選択（特徴量エンジニアリング），機械学習の手法の検討を行った経過を紹介する．用いるデータは，臨床試験のデータではなく，デバイスにより自動的に記録されたデータと，日誌など紙に記録されたデータの患者が混在しており，小規模な RWD である．

本章は，実際に機械学習を行うことの詳細な説明に重点を置いているため，第 1 章で紹介したフェーズを完全には踏襲していないが，おおむね次の構成となっている．

表　4 -1 本章の構成と第 1 章のプロジェクトフェーズとの対応

本章の見出し			第 1 章のフェーズ
第 1 節		実装例の目的	構想フェーズ
第 2 節	第 1 項	データの確認	
	第 2 項	分類モデルに適したデータ加工（特徴量の設定）	PoC フェーズ
	第 3 項	データの分割	
第 3 節	第 1 項	検討に用いる種々の教師あり学習手法	−
	第 2 項	手法と特徴量を検討して適切なものを選択する	PoC フェーズ
第 4 節	第 1 項	手法とパラメータの説明	−
	第 2 項	チューニング方法と評価指標	実装フェーズ
	第 3 項	パラメータのチューニングを行う	
	第 4 項	チューニングの結果	
第 5 節		実装	
第 7 節		深層学習による血糖値の予測	PoC フェーズ

第 1 節 実装例の目的

食前血糖値，インスリン投与量から低血糖イベントを予測するモデルを構築する．このモデルを用いてシミュレーションを行うことで，低血糖イベントを生じないインスリン投与量を推定する[注 23]．作成するサービスとしては，推定する対象の患者の数週間分のデータを用いて学習することで，患者さんごとのモデルを構築して使用する方法と，多くの患者のデータを用いて予め学習したモデルを用いる方法が考えられる．患者個々の様々な属性による影響が大きい場合は前者の精度が高くなると考えられるが，サービスを利用するためには，患者の長期のデータ蓄積が必要となる．後者は，患者の肥満度，性別年齢，運動

注23 この方法ではインスリン投与量の上限の推定となるため，インスリン投与量が不十分で血糖値が高くなる結果となることが予想される．そのため，本来は食後血糖値及び次の食前血糖値を用いて，インスリン投与量の下限値も推定する必要があるが，機械学習の事例紹介として簡便なものとするため，前者のみを行う．

量，食事量など詳細なデータにより患者の特徴を表現することができればデータの蓄積のない患者にも適用可能な汎用性のあるモデルの構築が可能である．

糖尿病は，インスリン作用不足による高血糖状態を主徴とする疾患である．一方で，治療薬の種類や量の誤り及び食事量が通常よりも少ない時などは低血糖を起こす場合がある．そこで，血糖値の推移の予測あるいは低血糖の生起を予測することは大変意義があると考えた．前者の血糖値の推移予測については，血糖値を時間経過とともに変動する時系列データとして扱い，直前までの変動パターンから次の時点の血糖値を推測するというモデルが利用できる．しかしながら，本データが食事前の空腹時血糖を中心としたスポットの値であることもあり，論文で報告されているような精度で血糖値の推移を予測することはできなかった（第 7 節参照）．

後者の低血糖の生起予測という二値分類の視点からは，血糖値は食事（糖質の量）とインスリン投与量に大きな影響を受け，これに運動などの活動状況が影響を与えるため，血糖値の変動ではなく，インスリン投与量とその時点の血糖値から低血糖を予測する分類モデルを扱うこととする．

第 2 節 データ

カリフォルニア大学アーバイン校の機械学習及びインテリジェントシステムセンターが運営している「UC Irvine Machine Learning Repository」に公開されている「Diabetes Data Set」[26]（https://archive.ics.uci.edu/ml/datasets/Diabetes）．1994 年 AAAI 春のシンポジウムで使用されたデータセット．日付，時分と測定項目名（インスリン使用量，朝食後血糖値，昼食後血糖値，食事量，運動量，低血糖イベント等 18 項目），値のテーブルになっており，時系列データの解析のサンプルデータである．データは，data-01 から data-70 までの 70 人の患者ごとのファイルである．データは tab 区切りのテキストファイルで，日付，時刻，データ項目コード，値の 4 列からなる．

表 4 -2 data ファイルのサンプル

04-21-1991	9:09	58	100
04-21-1991	9:09	33	009
04-21-1991	9:09	34	013
04-21-1991	17:08	62	119
04-21-1991	17:08	33	007
04-21-1991	22:51	48	123
04-22-1991	7:35	58	216

表４-3 データ項目コード

33 = Regular insulin dose
34 = NPH insulin dose
35 = UltraLente insulin dose
48 = Unspecified blood glucose measurement
57 = Unspecified blood glucose measurement
58 = Pre-breakfast blood glucose measurement
59 = Post-breakfast blood glucose measurement
60 = Pre-lunch blood glucose measurement
61 = Post-lunch blood glucose measurement
62 = Pre-supper blood glucose measurement
63 = Post-supper blood glucose measurement
64 = Pre-snack blood glucose measurement
65 = Hypoglycemic symptoms
66 = Typical meal ingestion
67 = More-than-usual meal ingestion
68 = Less-than-usual meal ingestion
69 = Typical exercise activity
70 = More-than-usual exercise activity
71 = Less-than-usual exercise activity
72 = Unspecified special event

第１項 データの確認

まず，データで用いられている値，コード，件数，欠測・欠損などを確認する．以後の作業を容易にするため，患者番号（patno）の列を追加して１ファイルに連結し，コードにラベルを設定する．すべて連結したデータは 29330 行である．

データが壊れていないか，抜け落ちていたり，間違って処理していないかといったことを確認する．日付にブランクとあり得ない６月 31 日が，項目コードにはマスタにない 0，4，36，56 があり，値にブランクが 51 件ある．６月 31 日は前後から６月 30 日と推測できるが，ブランクの日付とともに削除した．ブランク及びマスタにない項目コードのレコードは全て表４-4 のようになっており，インスリン投与量のレコードがずれていると思われるが，これも削除した．次に学習用の特徴量とターゲットに用いる変数を検討する前の確認として，項目ごとのレコード数，患者数と要約統計量（表４-5，表４-7），患者ごとの各項目のデータ数（表４-5，表４-6），日時の繰り返し数・期間（表４-7），患者ごとの食前血糖値の推移（図４-1）などを，機械学習では，ほとんどが同じ値である変数は特徴量として有用でない場合が多いことを念頭に確認する．数値の表だけでなく，図４-2 のようにヒストグラムなどによって視覚的に確認する．特に，予測のターゲットとなる低血糖イベントと血糖値については頻度，分布（図４-4，図４-6），タイミング（図４-1，図４-3），重要な特徴量であるインスリンの投与頻度と量（表４-6，図４-6）は多面的に確認する．

表４-4 ブランクのレコードの例

10-11-1989	23:00	48	Unspecified blood glucose measurement	127
10-12-1989	7:00	0		
		33	Regular insulin dose	
10-12-1989	7:00	0		
		33	Regular insulin dose	21
10-12-1989	7:00	0		
		33	Regular insulin dose	22
10-12-1989	11:30	60	Pre-lunch blood glucose measurement	169

表４-5 項目ごとのレコード数とデータがある患者数と要約統計量

データ項目	レコード数	患者数	行数	N	平均	標準偏差	最小値	最大値	欠測値N
33, Regular insulin dose	9518	65	9518	9507	6.6509	5.4588	0	344	11
34, NPH insulin dose	3830	59	3830	3830	15.394	14.155	1	388	0
35, UltraLente insulin dose	1053	19	1053	1053	15.858	7.8089	3	30	0
36			1	1	5	5	5	0	
48, Unspecified blood glucose measurement	1883	23	1883	1881	171.12	78.57	28	487	2
56			119	119	150.82	82.788	31	501	0
57, Unspecified blood glucose measurement	990	43	990	990	160.12	98.986	0	501	0
58, Pre-breakfast blood glucose measurement	3518	64	3518	3518	170.7	77.737	23	461	0
59, Post-breakfast blood glucose measurement	20	16	20	20	178.35	81.883	49	339	0
60, Pre-lunch blood glucose measurement	2771	60	2771	2768	143.56	74.559	15	452	3
61, Post-lunch blood glucose measurement	66	19	66	66	244.52	104.36	25	476	0
62, Pre-supper blood glucose measurement	3160	64	3160	3158	156.38	74.1	28	450	2
63, Post-supper blood glucose measurement	219	23	219	219	186.15	92.158	37	413	0
64, Pre-snack blood glucose measurement	904	42	904	904	148.77	86.732	0	461	0
65, Hypoglycemic symptoms	331	38	331	331	0	0	0	0	0
66, Typical meal ingestion	154	17	154	154	0	0	0	0	0
67, More-than-usual meal ingestion	326	35	326	326	0	0	0	0	0
68, Less-than-usual meal ingestion	34	17	34	34	0	0	0	0	0
69, Typical exercise activity	68	21	68	68	0	0	0	0	0
70, More-than-usual exercise activity	139	24	139	139	0	0	0	0	0
71, Less-than-usual exercise activity	98	25	98	98	0	0	0	0	0
72, Unspecified special event	94	31	94	94	0	0	0	0	0

表４-6 患者ごとの各項目のデータ数（一部，１列目の 1～6 は患者番号）

code	0	4	33	34	35	36	48	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
合計	33	1	9518	3830	1053	1	1883	119	990	3518	20	2771	66	3160	219	904	331	154	326	34	68	139	98	94
1	0	0	384	139	0	0	78	0	0	135	0	54	0	102	0	0	51	0	0	0	0	0	0	0
2	0	0	378	3	0	0	94	0	0	96	0	95	0	95	0	0	0	0	0	0	0	0	0	0
3	0	0	60	1	73	0	0	10	2	40	0	25	5	30	21	16	3	0	10	0	0	0	0	4
4	0	0	69	58	18	0	0	6	4	35	0	26	2	32	10	32	1	0	6	0	0	0	0	1
5	0	0	72	76	0	0	0	6	5	37	0	25	1	33	7	36	0	0	2	0	0	0	0	0
6	0	0	59	25	0	0	9	0	0	21	0	18	0	17	0	0	0	0	0	0	0	0	0	0

表４-7 患者ごとの期間等の確認（一部）

patno	合計(行数)	日時数	最小値(date-time)	最大値(date-time)	範囲(date)
1	943	510	1991/4/21 9:09	1991/9/3 7:20	135
2	761	380	1989/10/10 8:00	1990/1/13 12:00	95
3	300	229	1990/7/21 6:43	1990/8/28 16:55	38
4	300	226	1990/8/19 17:00	1990/9/26 17:20	38
5	300	227	1990/9/1 16:48	1990/10/11 17:34	40
6	149	70	1989/4/29 8:00	1989/5/23 8:00	24

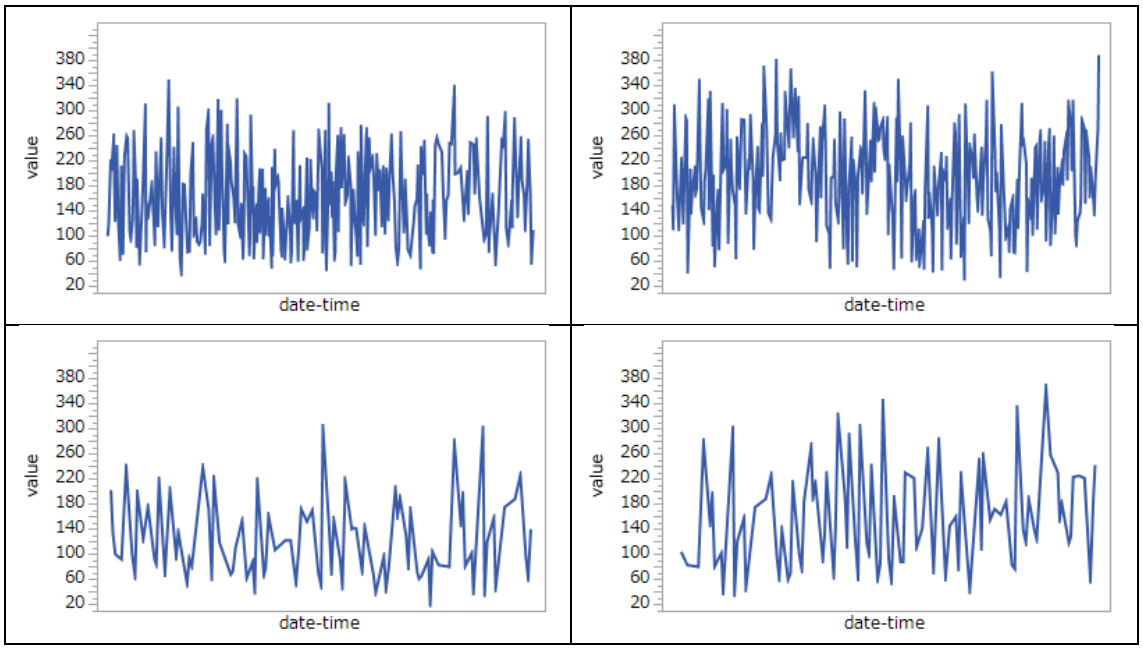


図４-1 患者ごと食前血糖値の推移の確認（一部）

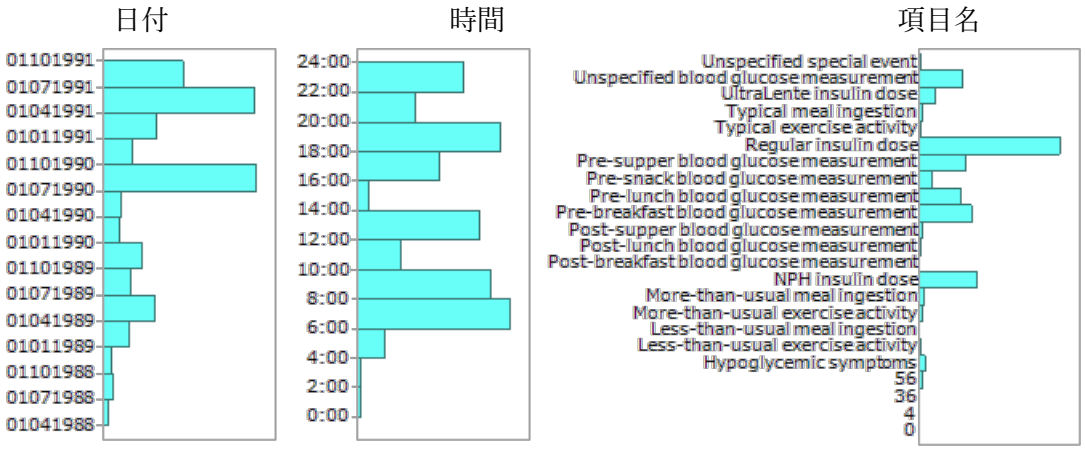


図４-2 日付，時間，項目名の頻度

低血糖イベントは 331 件，70 名中 38 名に記録されており，1 名の最大は 51 回である．患者ごとの期間（日数）とイベント発生状況は図４-3，図４-4，インスリン投与量，血糖値の分布は図４-6 となっている．更に詳細に見てゆくと，食後又は随意の血糖値が 70 以下であるが，低血糖イベントが記録されていないものが散見されるため，食後，随意及び次回の空腹時の血糖値が 70 以下を低血糖イベントありと修正すると患者ごとの頻度は図４-5 のようになり，低血糖イベントありは 2373 件，64 名となる．

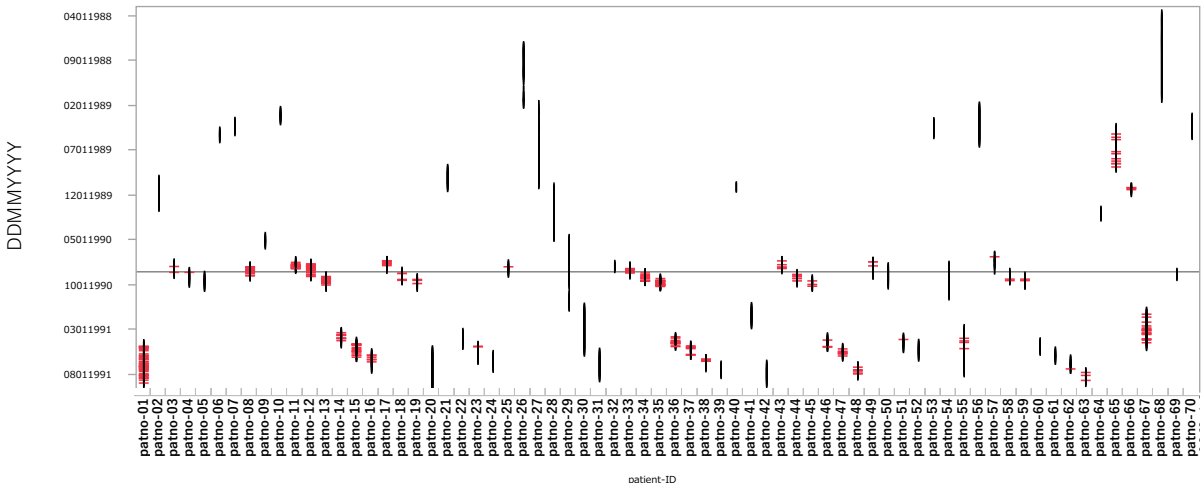


図４-3 患者ごとの日数分布（低血糖イベントを赤で表示）

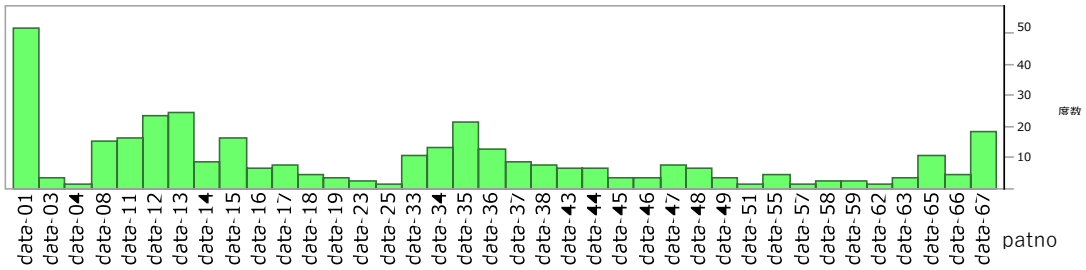


図４-4 患者ごとの低血糖イベント数

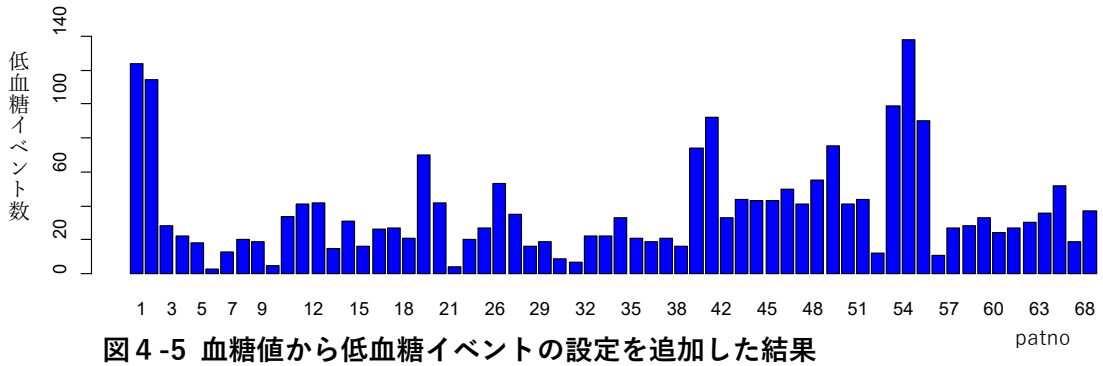


図４-5 血糖値から低血糖イベントの設定を追加した結果

図 4 -6 インスリン投与量，血糖値の分布

第 2 項 分類モデルに適したデータ加工（特徴量の設定）

この分類モデルにはいくつかの手法が利用可能であるが，いずれの場合でも，食前血糖値測定―インスリン投与－食事から次のインスリン投与直前までが 1 組の区間（「暴露イベント区間」とする）で低血糖イベントの有無を予測するシンプルなモデルとすることが可能で，それに合わせてデータの加工が必要となる．この各暴露イベント区間には順序，朝，昼，夜という時間帯もあるのでその変数（term）を加え，反対に，食事量，運動量のデータはほとんど記録されていないので除き，血糖値を対数変換した log_Pre_blood_glucose_measurement，直前 2 回を含む 3 回[注 24]の血糖値の単回帰の傾き：glucose_slp，同じ 3 回の血糖値の標準偏差：glucose_sd を追加して，表 4 -8 のようにデータを加工する．ここで NPH insulin，UltraLente insulin は約 24 時間作用するため，同じ日のレコードに投与量を 1/3 としてコピーしている．

注24 3 回の統計量以外に，5 回の統計量及び直前値との比の対数について検討を行い，3 回を選択した．

表 4 -8 加工後のデータの例

	patno	jyun	datetime	Hypoglycemic_symptoms	Hypoglycemic	UltraLente_insulin_dose	Regular_insulin_dose	NPH_insulin_dose	Low_MG	Pre_blood_glucose_measurement	term	glucose_slp	glucose_sd	UltraLente_NPH_insulin_dose	log_Pre_blood_glucose_measurement
329	1	346	682040700	0	0	0	8	16	0	199	morning	-5	43.589	16	0.298853076
331	1	348	682074000	0	0	0	7	0	0	140	nighttime	5.5	37.6431	0	0.146128036
332	1	349	682127400	0	0	0	9	15	0	249	morning	25	54.5619	15	0.396199347
336	1	354	682211700	0	0	0	9	15	0	246	morning	53	62.0833	15	0.390935107
338	1	356	682246800	0	0	0	6	0	0	173	nighttime	-38	43.0387	0	0.238046103
339	1	357	682290900	1	1	0	9	15	0	251	morning	2.5	43.6616	15	0.399673721
343	1	361	682382700	0	0	0	8	15	0	159	morning	-7	49.5715	15	0.201397124
347	1	365	682506000	0	0	0	4	0	0	95	nighttime	-78	78.4177	0	-0.022276395
348	1	366	682558200	0	0	0	8	15	0	103	morning	-28	34.8712	15	0.012837225
350	1	368	682597800	0	0	0	7	0	0	285	nighttime	95	107.462	0	0.45484486
351	1	369	682635360	0	0	0	8	15	0	98	morning	-2.5	106.55	15	-0.008773924
353	1	371	682733700	0	0	0	4	15	0	163	morning	-61	94.9368	15	0.212187604
355	1	373	682816920	0	0	0	8	15	0	59	morning	-19.5	52.5389	15	-0.229147988
358	1	377	682896300	0	0	0	8	15	0	138	morning	-12.5	54.2863	15	0.139879086

ここまでの，表 4 -5 及び表 4 -8 から，空腹時血糖値がない，インスリンの投与が全くないという問題があるレコードが生じていることに気づいた．インスリンの投与がない場合は投与されていない，すなわち「0」とすることが正しく，計算上問題ないが，空腹時血糖値は測定・記録漏れであり，欠損とするあるいは何らかの仮定を用いて補完する必要がある．表 4 -9 により，患者・低血糖イベントの有無別に欠損件数を確認すると患者ごとでばらつきがあるが，全体では低血糖イベントなしのレコード 11264 件の 13%，ありのレコード 470 件の 16%に欠測がある．今回のモデルは食前の空腹時血糖値の測定を条件としているので，欠測レコードは除外することとする．ここまでの処理を行った上で，データの少ない患者（10 件未満），低血糖イベントが 2 件以下の患者を除く．作成されたデータは念のため要約統計量，箱ひげ図，散布図行列などをもちいて確認する（表 4 -10，図 4 -7，図 4 -8）．元の低血糖イベント（Hypoglycemic）は 10053 レコード中 267 件で 2.7%に過ぎないが，血糖値が 70 未満がある場合を加えると 2373 件 24%となる．図 4 -8 から低血糖イベントの有無間で空腹時血糖値と直近の傾向（3 回の傾き），Regular_insulin_dose，UltraLente_insulin_dose は僅かながら差が生じているかもしれない．

表 4 -9 患者ごとの空腹時血糖の欠測レコード数

	低血糖イベント	
patno	0	1
1	88	15
2	97	2
3	9	1
4	9	0
5	7	0
6	7	0
7	7	1
8	2	2
9	9	0
10	4	0
11	6	0
12	7	0
13	10	0
14	5	0
15	4	0
16	4	0
17	3	2
18	7	0

	低血糖イベント	
patno	0	1
19	5	0
20	20	0
21	45	0
22	98	0
23	93	2
24	40	0
25	12	1
26	0	0
27	6	0
28	2	0
29	1	0
30	0	0
31	1	0
32	5	0
33	12	1
34	10	0
35	17	0
36	8	2

	低血糖イベント	
patno	0	1
37	8	0
38	7	0
39	2	0
40	32	0
41	64	3
42	64	5
43	28	4
44	36	5
45	38	2
46	38	3
47	32	5
48	39	6
49	0	1
50	0	0
51	3	1
52	1	0
53	3	1

	低血糖イベント	
patno	0	1
54	55	0
55	114	3
56	26	1
57	8	0
58	21	0
59	22	0
60	10	0
61	11	0
62	12	0
63	18	0
64	1	0
65	11	0
66	57	4
67	13	0
68	1	0
69	1	0
70	40	2
計	1476	75

表 4 -10 データの要約（統計量） R の summary(データセット)で出力される

```
patno           jyun          datetime          Hypoglycemic_symptoms Hypoglycemic UltraLente_insulin_dose Regular_insulin_dose NPH_insulin_dose
68     : 486   Min.   :  3.0   Min.   :575456400   0:7680                0:9786       Min.   : 0.000         Min.   : 0.000       Min.   :  0.000
29     : 453   1st Qu.: 42.0   1st Qu.:628142400   1:2373                1: 267       1st Qu.: 0.000         1st Qu.: 2.000       1st Qu.:  0.000
55     : 445   Median : 87.0   Median :653384280                                      Median : 0.000         Median : 5.000       Median :  0.000
30     : 439   Mean   :126.2   Mean   :649066301                                      Mean   : 3.739         Mean   : 5.251       Mean   :  5.007
65     : 390   3rd Qu.:188.0   3rd Qu.:673235100                                      3rd Qu.: 0.000         3rd Qu.: 7.000       3rd Qu.:  7.000
20     : 388   Max.   :488.0   Max.   :685613700                                      Max.   :30.000         Max.   :60.000       Max.   :296.000
(Other):7452
     Low_MG       Pre_blood_glucose_measurement        term      UltraLente_NPH_insulin_dose log_Pre_blood_glucose_measurement  glucose_slp
Min.   :  0.00   Min.   : 15.0                 daytime  :2703   Min.   :  0.000             Min.   :-0.82391                  Min.   :-211.00000
1st Qu.:  0.00   1st Qu.: 96.0                 morning  :3511   1st Qu.:  0.000             1st Qu.:-0.01773                  1st Qu.: -29.00000
Median :  0.00   Median :146.0                 nighttime:3839   Median :  2.000             Median : 0.16435                  Median :   0.00000
Mean   : 30.46   Mean   :157.1                                  Mean   :  8.745             Mean   : 0.13972                  Mean   :  -0.06665
3rd Qu.: 35.00   3rd Qu.:205.0                                  3rd Qu.: 16.000             3rd Qu.: 0.31175                  3rd Qu.:  29.00000
Max.   :501.00   Max.   :461.0                                  Max.   :326.000             Max.   : 0.66370                  Max.   : 191.00000

  glucose_sd      glucose_ave      pglucose_sd      pglucose_ave     pRegular_ave      pNPH_ave      pUltraLente_ave
Min.   :  0.5774   Min.   : 42.0   Min.   :  0.00   Min.   :  0.0   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
1st Qu.: 28.3608   1st Qu.:119.0   1st Qu.: 44.36   1st Qu.:131.7   1st Qu.: 2.200   1st Qu.: 0.000   1st Qu.: 0.000
Median : 51.7301   Median :151.3   Median : 65.68   Median :161.2   Median : 5.400   Median : 6.400   Median : 0.000
Mean   : 59.1530   Mean   :157.2   Mean   : 65.36   Mean   :163.3   Mean   : 5.124   Mean   : 5.929   Mean   : 2.579
3rd Qu.: 83.1625   3rd Qu.:190.7   3rd Qu.: 80.18   3rd Qu.:196.1   3rd Qu.: 6.500   3rd Qu.: 9.200   3rd Qu.: 0.000
Max.   :235.4598   Max.   :372.3   Max.   :119.38   Max.   :293.0   Max.   :14.400   Max.   :23.000   Max.   :24.400
```

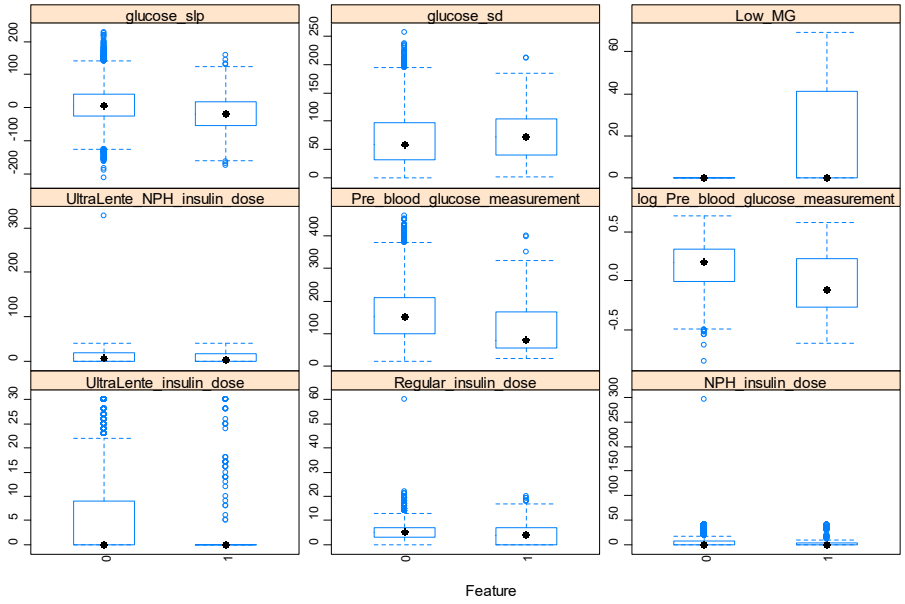


図 4 -7 連続変数データの低血糖有無別の Box plots

図 4 -8 作成した機械学習データの 2 変数ごとの概要（低血糖イベント別）

後述する学習データを用いてモデルを

Hypoglycemic_symptoms ~ UltraLente_insulin_dose + Regular_insulin_dose + NPH_insulin_dose +UltraLente_NPH_insulin_dose + log_Pre_blood_glucose_measurement + glucose_slp + glucose_sd + term + as.factor(patno)

として単純なロジスティック解析を行うと時間帯（term）の影響は大きく，インスリン投与量，空腹時血糖値との関連は小さいようである．また，このモデルによる識別の混同行列（表 4 -12）から，正解率は 0.783 であるが，再現率は 0.207，特異度は 0.962 である．低血糖の予測としては不十分であるので，機械学習による精度向上を試みる．

表 4 -11 単純なロジスティックモデルによる各要因のオッズ比

	OR	OR95%CI
(Intercept)	0.509	0.345 ～ 0.750
UltraLente_insulin_dose	0.988	0.959 ～ 1.018
Regular_insulin_dose	1.126	1.095 ～ 1.159
NPH_insulin_dose	0.987	0.977 ～ 0.998
log_Pre_blood_glucose_measurement	0.234	0.160 ～ 0.343
glucose_slp	1.001	1.000 ～ 1.003
glucose_sd	0.999	0.997 ～ 1.001
termdaytime	0.656	0.536 ～ 0.804
termnighttime	1.146	0.967 ～ 1.357
mean patno	0.956	0.483 ～ 1.914

表 4 -12 単純なロジスティックモデルによるテスト結果

		予測		特異度/再現率
		0	1	
観察	0	2452	96	0.962
	1	629	164	0.207

正解率： 0.7830

第 3 項 データの分割

機械学習を始める前にデータを学習（開発）用とテスト用に分ける．この糖尿病データは，識別対象の低血糖イベントが少なく，患者に関する情報を特徴量に用いるため，低血糖イベント及び患者の分布が偏らないように，低血糖イベントと患者の層別に開発（学習）用とテスト用に 2 対 1 に分ける[注 25]．学習結果の評価にはホールドアウト法を用いるが，学習中の学習データと検証データの分割はブートストラップ法を用いた．

表 4 -13 分割したデータの患者ごと低血糖有無別頻度（%，一部抜粋）

patno	テスト		学習	
	低血糖あり	低血糖なし	低血糖あり	低血糖なし
1	43	57	45	55
2	39	61	46	54
3	22	78	34	66
・・・・・				
68	5	95	2	98
70	39	61	52	48
総計	23	77	24	76

注25 実際には，この後で患者ごとの最小の 10 件のデータを特徴量に用いるため，その 10 件は全て学習データ，テストデータ間で重複しないように処理している．イメージは第 3 章第 1 節第 3 項参照

第 3 節 手法（アルゴリズム）

第 1 項 検討に用いる種々の教師あり学習手法

二値分類であるので，基本的な機械学習の手法及びその改良された多くの手法が利用可能である．決定木である CART (rpart) [注 26]，k 近傍法である k-Nearest Neighbors (knn)，ナイーブベイズである Naive Bayes（nb），Learning Vector Quantization（lvq），サポートベクターマシンである Support Vector Machines with Radial Basis Function Kernel（svmRadial)，ブースティングの Random Forest(rf)，Stochastic Gradient Boosting(gbm)，ニューラルネットワークの Neural Network（nnet），勾配ブースティングの eXtreme Gradient Boosting（xgbTree）の 9 種類を用いた．
これらは基本的な手法を改良又は組み合わせたものであるので，基本的手法の概略を説明した後に，各手法の概要を示す．

A） 基本的手法

決定木

決定木（decision tree）とは学習データを条件分岐によって分割し，木構造を用いて分類や回帰を行う機械学習の手法の 1 つである．分割する際には，ジニ係数やエントロピーが用いられる．対象の数値を推定する問題を解く回帰木(regression tree)と対象を分類する問題を解く分類木（classification tree)の総称である．

k 近傍法

k 近傍法（k-nearest neighbor algorithm, k-NN）は，2 つ以上の多クラスを含む分類問題を解くのに利用される機械学習の手法の 1 つである．k 近傍法は他の機械学習アルゴリズムとは異なり，学習データからモデルのパラメータ推測するというステップが存在せず，未知データの分類時には，学習データとの距離を計算し，近傍の k 個の点がどちらのクラスに属するか多数決にて分類する．距離の計算には一般にはユークリッド距離が用いられるが，それ以外の計算方法や重みづけするものもある．入力データと学習データ中の全サンプルとの距離を計算し，入力に近い方から k 個の学習データを取得する．学習データのラベルで多数決を行い，最も多いクラスをテストデータのサンプルが属するクラスに分類する．

ニューラルネットワーク

ニューラルネットワーク（Neural Network：NN）は人間の脳内にある神経回路網を人工ニューロンという数式的なモデルで表現したもので，深層学習（ディープラーニング）の基礎となる学習器[注 27]である．1 または多数のニューロンが並んだ入力層，隠れ層（通常は 1 層），出力層の 3 層により構成され，層と層の間にはニューロン同士のつながりの強さを示

注26 以下，一般的な手法名と（）内は R caret パッケージで用いる名称である．
https://cran.r-project.org/web/packages/caret/caret.pdf

注27 学習器は，機械学習によって作られた識別，予測などを行える学習モデルである．手法の説明では「弱学習器」という用語を用いるため，ここでは，揃えて学習器と表記している．

す重みが学習により調整される．個々の人工ニューロンは単純な仕組み（シグモイド関数などの活性化関数）であるが，それを多数組み合わせる事で複雑な関数近似が可能である．

ナイーブベイズ

ナイーブベイズ（Naive Bayes；単純ベイズ）は下に示したベイズの定理を用いた分類問題を解く機械学習の手法である．このアルゴリズムでは，ある入力データが与えられたときに特定のクラスに分類されるすべての出力の推定確率を算出し，その中で，最も確率の高いものを推定結果とする．ここで，データの特徴量は独立かつ互いに相関がないといった仮定が必要になるため，タスクによっては，この仮定が成立しない場合もある．しかしながら，こうした強い仮定が成り立たないであろう実データに対しても良い結果を出す場合があり，これがナイーブベイズという名称の所以でもある．

$$\text{ベイズの定理：}\mathrm{P}(H|D) = \frac{P(D|H)P(H)}{P(D)}$$

$\mathrm{P}(H|D)$：事後確率（データ D が与えられたときの推定 H が正しい確率）
$P(D|H)$：原因の推定 H が正しいとしたときの結果 D の確率
$P(H)$：推定 H の事前確率（結果 D に関係なく原因の推定 H が正しい確率）
$P(D)$：データ D の事前確率（原因の推定に関係ない結果 D の確率）

例えば，インフルエンザに罹患する確率$P(H) = 0.05$，体温が 38℃以上である確率$P(D) = 0.07$，インフルエンザに罹患して体温が 38℃以上となる確率を$P(D|H) = 0.8$とすると，体温が 38℃以上の場合にインフルエンザに罹患している確率は

$$P(H|D) = (0.8 \times 0.05)/0.07 = 0.57$$

となる．この場合はインフルエンザである/ないの二値であるが，ナイーブベイズは，他の様々な感染症（RS ウイルス，コロナウイルス，ノロウイルス，おたふくかぜ etc.）の罹患についても同様の方法で推定の確率$P(H|D)$を計算し，その中で最も確率の高いもの（分類，クラス）を推定結果として返す．

サポートベクターマシーン

クラスの決定境界線を求めることで分類を行う線形識別器である．正しい決定境界線を見つけるために，判別する境界とデータとの距離が最大となるようにする「マージン最大化」が用いられる．境界線は本来直線であるが，カーネル関数を用いることで，高次元空間からの投影により非線形な決定境界による識別も行える．

アンサンブル学習

アンサンブル学習は複数の手法・モデルを用いて，それらの予測結果を統合して汎化性能を高める方法である．個々の性能は良くない学習器（弱学習器)を複数束ねることで性能の良い学習器が得られる．一般的に，分類問題では多数決，回帰問題では平均，中央値などが出力となる．アンサンブル学習は学習データをランダムに復元抽出することで多様性を持

たせられるバギング（Bootstrap Aggregating），複数のモデルの予測精度に応じた重みでの調整（Adaptive Boost)や誤差を修正するようにモデルを更新する(Gradient Boost)といったブースティング（Boosting），ロジスティック回帰やランダムフォレストなど様々な学習器による予測値を用いて更に正解率が高い学習器の組合せが選ばれるスタッキング（Stacking）に分けられる．

バギング

アンサンブル学習の 1 つで，学習データをブートストラップサンプリング（復元抽出）により多数作成し，独立に多数の弱学習器を作って「多数決を取ったものを出力とする」といった手法である．次のブースティングとは異なり，同時に弱学習器を作ることができるので，並列計算ができる環境では高速に学習が可能である．ランダムフォレストはバギングを用いる代表例である．

ブースティング

ブースティングはバギングのように弱学習器を独立に作るのではなく，1 つずつ順番に前の学習器の弱点（分類の誤り）を補完する．勾配ブースティングは，誤差を修正するようにモデルを更新して強学習器を構築する．アダプティブブースティングは，正しく分類されたサンプルと誤分類されたサンプルについて，重みを変え，反復の度に再び重みづけを更新することで，学習済みの弱学習器の誤答から強学習器を構築する．

スタッキング

スタッキングは 1 段目の弱学習器（モデル）を使った予測値を特徴量に加えて 2 段目のモデルを作ることを繰り返すアンサンブル学習の手法である．つまり，1 段ごとに特徴量が 1 つ増加する．1 段目のモデルはバギングのように複数用いることもできる．

B） 検討に用いた手法

・CART（rpart）

CART (Classification and Regression Trees）は 1984 年に Breiman らによって考案された学習アルゴリズムである．CART は分類及び回帰問題の両方に対応する．CART では説明変数を 2 進分岐させ，決定木を生成する．分岐の評価基準には不純度（impurity）を数値化したジニ係数（Gini index，またはジニ分散指標（Gini diversity index））を用い，変数の分割前後の差分で評価する．説明変数に対する分岐後の不純度を算出し，これらの中でも最も大きな値が得られる特徴量を優先して分岐条件に使用する．

・k-Nearest Neighbors（knn）

シンプルな k 近傍法であり，計算には一般的なユークリッド距離（L_2 距離）のほか，マンハッタン距離（L_1 距離），マハラノビス距離などが用いられる．

・Naive Bayes（nb）

シンプルなナイーブベイズ法であるが，一般的なガウス（Gauss）分布以外にも近似と

してカーネル密度（kernel-density）も利用可能である．

・Learning Vector Quantization（lvq）

ベクトル量子化（Vector Quantization）とはベクトルで表されたデータ集合を有限個の代表的なパターンに近似し置き換える処理のことであり，学習ベクトル量子化（Learning Vector Quantization）はベクトル量子化の教師あり機械学習法を指す．学習ベクトル量子化のアルゴリズムでは各クラスの特徴空間にコードブックと呼ばれる参照点を定める．このコードブックは各クラスに複数個設定される．データのサンプルが入力されると，最もユークリッド距離が近いコードブックを算出し，そのコードブックに対応したクラスにデータサンプルを分類する．学習方法の違いにより，LVQ1, LVQ2, LVQ3 及び OLVQ1 といった方法が提案されている．

・Support Vector Machines with Radial Basis Function Kernel（svmRadial）

サポートベクターマシーン（SVM）は分類及び回帰問題に適応される機械学習法である．初期の SVM は線形分離可能な問題にのみ対応していたが，その後カーネル関数を用いることで非線形な問題にも対応が可能となった．カーネル関数には，線形カーネル，シグモイドカーネル，多項カーネルなどがあるが，SVM においては RBF カーネル（Radial Basis Function Kernel）が用いられることが多く，これはガウス（ガウシアン）カーネルとも呼ばれる．

・Random Forest（rf）

ランダムフォレストは，2001 年に Leo Breiman によって提案された機械学習のアルゴリズムであり，分類，回帰，クラスタリングに適応される．決定木を弱学習器とするアンサンブル学習とみなすことができる．学習データを復元抽出（ブートストラップサンプリング）したデータセット[注 28]を用いて多数の決定木を作成する．その際に，一部の特徴量だけをランダムに学習させる．そして，それぞれの決定木から得られる予測結果の多数決を採ることで最終的な分類を行う．

・Gradient Boosting Machine（gbm）

gbm は，ブースティングの 1 つで，確率的勾配ブースティング（Stochastic Gradient Boosting）を行う．勾配ブースティングは，複数の弱学習器を組み合わせて全体で良い性能を得るアンサンブル学習の一種であり，その中でも 1 つずつ順番に，1 つ前の決定木を次の決定木で修正するように，弱学習器を構築していく手法である．そのため，並列計算ができず，学習に時間を要するが，過学習（高バリアンス）と学習不足（高バイアス）のバランスを取りやすい[注 29]．gbm は学習器に決定木を用い，弱学習器とするため，各決定木には浅いものが用いられる．具体的には，ステップごとに浅い決定木を構築して，前のステップにお

注28 例えば，1 万件のデータから復元抽出で 1 万件を抽出すると，複数回抽出されるデータが生まれるため僅かずつ異なるデータセットが作られる．

注29 順番に学習するのでなく，並列（独立）に学習する手法はバギングと呼ばれる．

ける間違った識別に対して損失関数が小さくなるように重みの調整を繰り返す．個々の決定木がどの程度補正を行うのかの強さは一般に学習率（learning_rate）をパラメータとして与えることで調整される．学習率は大きいほど補正が強い＝モデルが複雑になる．また，gbm は計算コストが高いため，LightBGM などいくつか改良された手法も提供されている．

・Neural Network（nnet）

シンプルなニューラルネットワーク（Neural Network）法であり，入力層，隠れ層，結合層（出力層）の 3 層からなる．オプションで入力と出力を直接結合して完全な単層とすることも可能である．計算方法は対数確率モデルを使用する entrop（最大条件付の尤度）と softmax（対数確率モデル）を使用できる．R には nnet のほかに，多層の隠れ層を用いることができる neuralnet ライブラリがある．

・eXtreme Gradient Boosting（xgbTree）

eXtreme Gradient Boosting（一般的な略称：XGBoost）は勾配ブースティング（Gradient Boosting)と Random Forest を組み合わせた手法の一つである．学習の任意の時点 t において，モデルの予測は t-1 時点の結果に基づき重み付けされる．正しく予測されたデータに対しては重みを小さくし，誤って予測されたデータに対しては重みを大きくすることで，後で学習する識別器ほど誤ったデータに集中して学習を進めることが可能となる．XGBoost は特に分類，回帰，そしてランク付けの問題に対し良いパフォーマンスを発揮する．

第 2 項 手法と特徴量を検討して適切なものを選択する

A） 分類モデルの評価指標

学習に用いる特徴量，手法及びモデルに用いるパラメータを比較評価するには統一した指標が必要である．機械学習の性能の評価には，大きく分けて学習時間，判定時間及び判定精度の 3 つがあるが，ここでは，判定精度に焦点を当て，分類モデルでは一般的な 4 つの指標（正解率，再現率，特異度，適合率）について紹介する．

分類モデルによる判定結果 / 観測データのラベル	低血糖なし	低血糖あり
低血糖なし	a	b
低血糖あり	c	d

- 正解率（accuracy）：分類モデル による判定結果と観測データのラベルが一致（ともに「低血糖なし」または「低血糖あり」）した割合

$$\text{正解率} = \frac{a+d}{a+b+c+d}$$

- 再現率（recall，感度：sensitivity，真陽性率：True Positive Rate）：観測データのラベルが「低血糖あり」であったデータのうち，学習器により「低血糖あり」と判定された割合

$$再現率 = \frac{d}{c+d}$$

- 特異度（specificity）：観測データのラベルが「低血糖なし」であったデータのうち，学習器により「低血糖なし」と判定された割合

$$特異度 = \frac{a}{a+b}$$

- 適合率（precision，陽性的中率：ppv）：学習器により「低血糖あり」と判定されたデータのうち，観測データのラベルも「低血糖あり」であった割合

$$適合率 = \frac{d}{b+d}$$

- F 尺度，（F-measure，F 値）：適合率及び再現率の調和平均

$$\mathrm{F} = \frac{2 \times \left(\frac{d}{b+d} \times \frac{d}{c+d}\right)}{\frac{d}{b+d} + \frac{d}{c+d}}$$

- AUC（Area Under the Curve）：ROC（Receiver Operating Characteristic）曲線（推測曲線）の右下の面積．0～1 の値をとる．ROC は，縦軸に真陽性率（再現率），横軸に偽陽性率（False Positive Rate，「低血糖なし」のラベルを誤って「低血糖あり」と判定した割合）を用いて，「推定確率が p 以上のものを低血糖ありと判定する」とした場合の p を 0～1 に変化させた曲線をプロットしたもの．

B） 観測データを用いて複数の手法を比較

イベントありは全体の約 24%と少ないため重みを 2 とした．分類モデルは患者を要因とする Hypoglycemic_symptoms ~ UltraLente_insulin_dose + Regular_insulin_dose + NPH_insulin_dose +Pre_blood_glucose_measurement + term +as.factor(patno)（model1）として，学習中の各パラメータは 5 パターン，抽出法は Bootstrapping により 25 回繰り返し（デフォルト），k-Nearest Neighbors（knn），CART（rpart），Learning Vector Quantization（lvq），Support Vector Machines with Radial Basis Function Kernel（svmRadial），Stochastic Gradient Boosting（gbm），Random Forest（rf），Neural Network（nnet），Naive Bayes（nb），eXtreme Gradient Boosting（xgbTree）の 9 種類を行った結果，学習は accuracy を指標とする場合，rf が最も精度が高く，accuracy=0.994 であった．テストデータでは accuracy は rf の 0.822，再現率は xgbTree の 0.467 が最良であったが，rf，xgbTree ともに学習に約 1 時間を要しており実用上問題があると思われる．

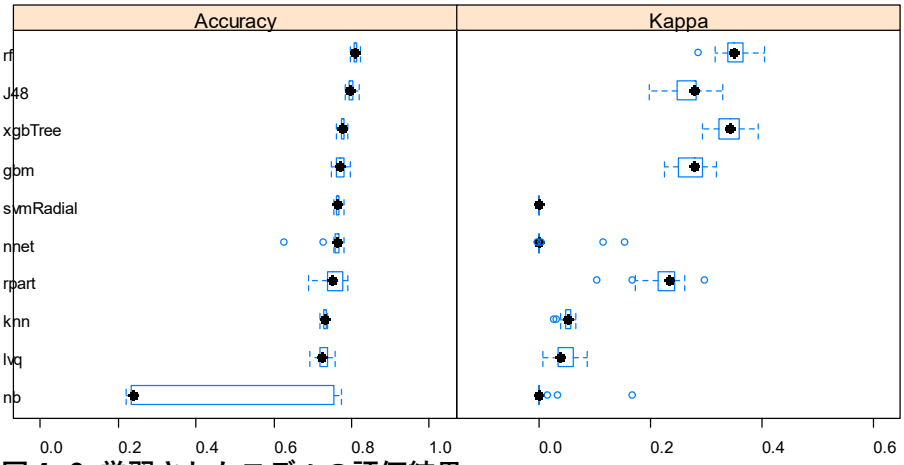


図 4 -9 学習されたモデルの評価結果

表 4 -14 学習の要約

	accuracy	recall	specificity	学習時間（分）	The final values used for the model
knn	0.7734	0.1144	0.9754	3.5	k=13
rpart	0.7900	0.3687	0.9191	0.1	cp = 0.005721551
lvq	0.7306	0.1742	0.9012	36.8	size = 30 and k = 21
svmRadial	0.7654	0	1	10.1	sigma(constant) = 0.0002278892 and C = 0.25
gbm	0.7851	0.3617	0.9148	5.3	n.trees = 50, interaction.depth = 4, shrinkage(constant) = 0.1 and n.minobsinnode(constant) = 10
rf	0.9937	0.9752	0.9994	57.8	mtry = 19
nnet	0.7209	0.1678	0.8905	6.1	size = 1 and decay = 1e-04
nb	0.2346	1	0	10.9	fL(constant) = 0, usekernel = TRUE and adjust(constant) = 1
xgbTree	0.9879	0.9828	0.9895	65.3	nrounds = 250, max_depth = 5, eta = 0.4, gamma(constant) = 0, colsample_bytree = 0.8, min_child_weight(constant) = 1 and subsample = 0.875

表 4 -15 各手法のテストデータの結果（NaN は 0 による除算により計算不能）

methods	accuracy	recall	specificity	precision	confusion table			
					cel00	cel01	cel10	cel11
knn	0.7579	0.0706	0.9717	0.4375	2476	72	737	56
rpart	0.7863	0.3607	0.9188	0.5801	2341	207	507	286
lvq	0.7157	0.1526	0.8909	0.3033	2270	278	672	121
svmRadial	0.7626	0	1	NaN	2548	0	793	0
gbm	0.7896	0.3670	0.9211	0.5915	2347	201	502	291
rf	0.8216	0.3607	0.9651	0.7627	2459	89	507	286
nnet	0.7210	0.1677	0.8932	0.3284	2276	272	660	133
nb	0.2374	1	0	0.2374	0	2548	0	793
xgbTree	0.7866	0.4666	0.8862	0.5606	2258	290	423	370

rf について詳細に確認すると，変数の寄与は図 4 -12 の通りで，特定の患者の影響が表れているのではなく，血糖値の変動（傾きと標準偏差），血糖値，インスリン使用量が寄与しているので歪なモデルではないようである．tree の数とエラー率の関係（図 4 -10）を見ると，低血糖イベントありのエラー率が高く，30 程度でピークに達するが，イベントなしのエラー率が減少するため，19 が選ばれたことが確認できる．また，寄与の大きい log_Pre_blood_glucose_measurement と glucose_slp の散布図に決定境界（緑の囲み）を表示すると図 4 -11 のように境界は複雑でうまく分離できていないことが確認できる．

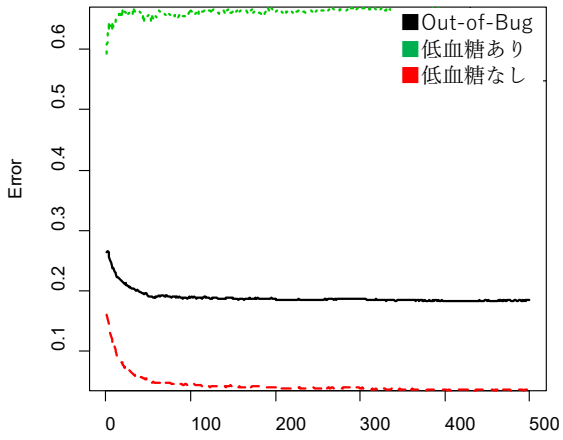


図 4 -10 rf の tree の数とエラー率の関係

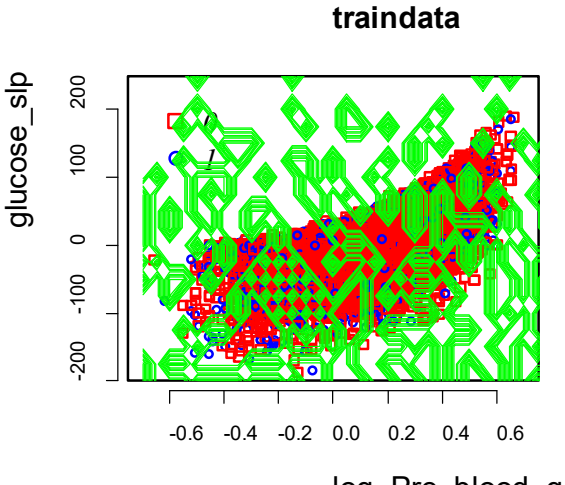


図 4 -11 決定境界（緑の囲み）

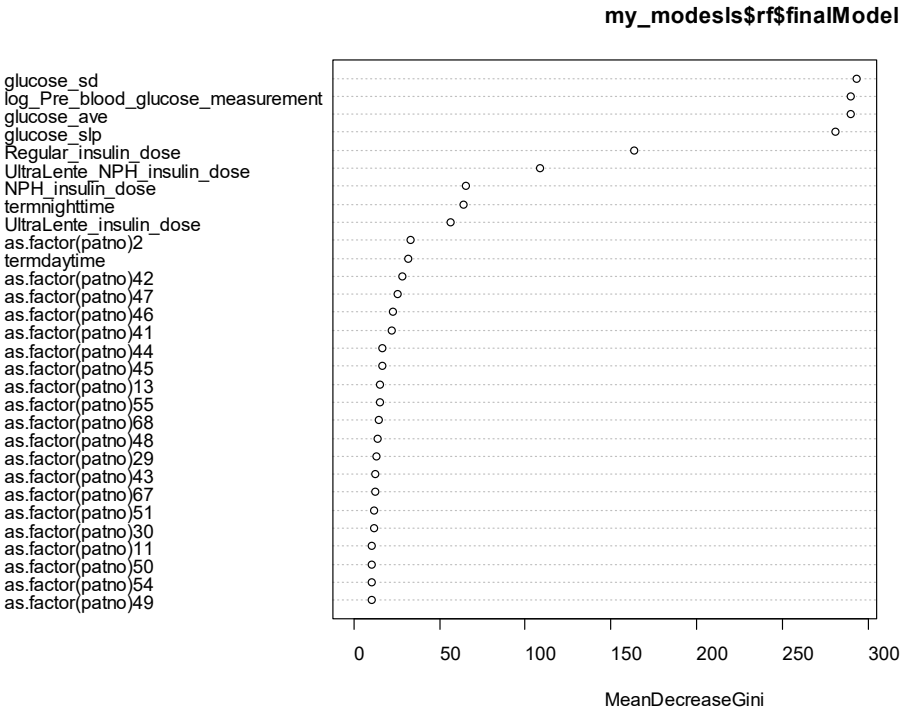


図 4 -12 rf の寄与した変数の視覚化

C） 合成による特徴量

ここまでは，観測されたデータをほぼそのまま用いてきたが，いずれの手法でも再現率は不十分である．ハイパーパラメータや特徴量の吟味により多少の改善は期待できるが，不十分かもしれない．そこで，主成分分析により新たな特徴量を抽出することを試みる．

学習データのインスリン使用ごとの各レコードの特徴量として連続変数である 7 変数（UltraLente_insulin_dose，Regular_insulin_dose，NPH_insulin_dose，UltraLente_NPH_insulin_dose，Pre_blood_glucose_measurement，glucose_slp，glucose_sd）を用いて主成分分析を行った（この主成分分析の関数は実装で利用するため保存する）[注 30]．各因子の分布は図 4 -13，各変数の固有ベクトルは表 4 -16 のとおりである．図 4 -14 の因子の散布図行列（右上は低血糖イベントの有無で色分け，左下は患者で色分け）から，低血糖イベントはいずれの因子の組合せにも集中している領域が認められないが，左下のプロットではいずれの組合せでも同じ色が並んでいるものが散見され，患者ごとに特徴が表れている．学習データに主成分分析の結果をそのまま用いると，各レコードが一意に特定される場合が多くなるため，値を 2 桁程度に丸めてデータに追加した Regular_insulin_dose，log_Pre_blood_glucose_measurement，glucose_slp，

注30 実装では，予測する対象患者のデータを用いた主成分分析を行わないため，テストデータは使用せず，学習データのみを用いている．

glucose_sd，PC1，PC2，PC3，PC4，PC5，PC6，PC7[注31]，term，patno（model2）により再度，9 種類の手法で学習を行った結果（表 4 -17，表 4 -18），model1 同様，正解率は rf の 0.821 が最良で，再現率は xgbTree の 0.479 が最良であった．rf は学習データでは再現率，特異度ともに 1 であり過学習を生じているが，gbm は学習データとテストデータで正解率，再現率とも差は小さく，gbm を用いるのが良さそうである．また，特徴量（変数）を増やしたため，学習においてユニークなデータがある警告がいくつか生じており，用いる変数を吟味して絞り込むことが必要である．

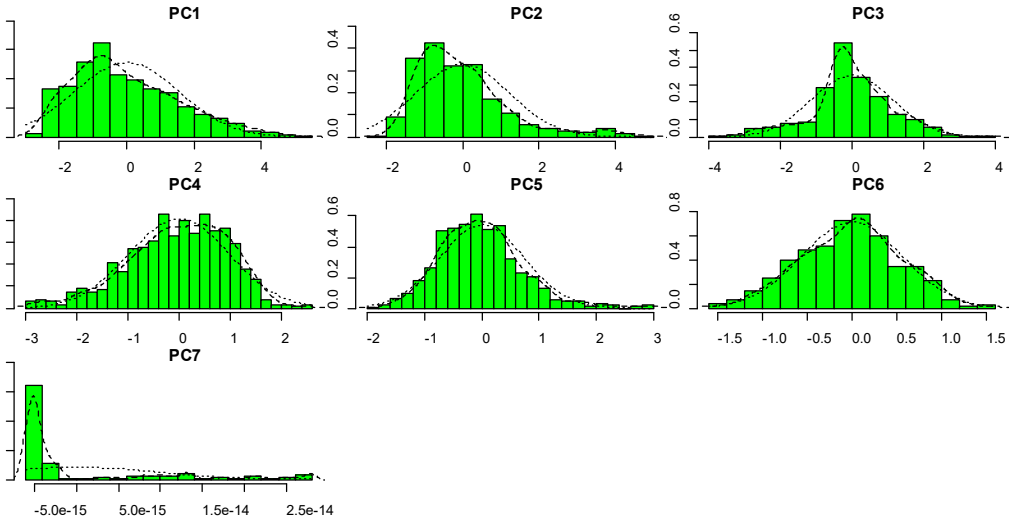


図 4 -13 主成分分析の各因子の分布

表 4 -16 固有ベクトル

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
UltraLente_insulin_dose	0.36833	-0.21547	-0.60046	-0.28108	-0.34900	0.12371	-0.49116
Regular_insulin_dose	0.49851	-0.05569	-0.22690	0.16115	0.63980	-0.51146	0.00000
NPH_insulin_dose	0.28114	0.63154	0.34568	0.24444	-0.04328	0.07478	-0.57914
log_Pre_blood_glucose_measurement	0.39151	-0.44458	0.33429	0.07255	0.29116	0.66879	0.00000
glucose_slp	0.30437	-0.41035	0.49344	0.07165	-0.50431	-0.48583	0.00000
glucose_sd	0.10855	0.16126	0.30338	-0.90821	0.19157	-0.09286	0.00000
UltraLente_NPH_insulin_dose	0.52828	0.39947	-0.14558	0.00539	-0.30197	0.15994	0.65066

注31 PC1 から PC7 は主成分分析の第 1 因子から第 7 因子

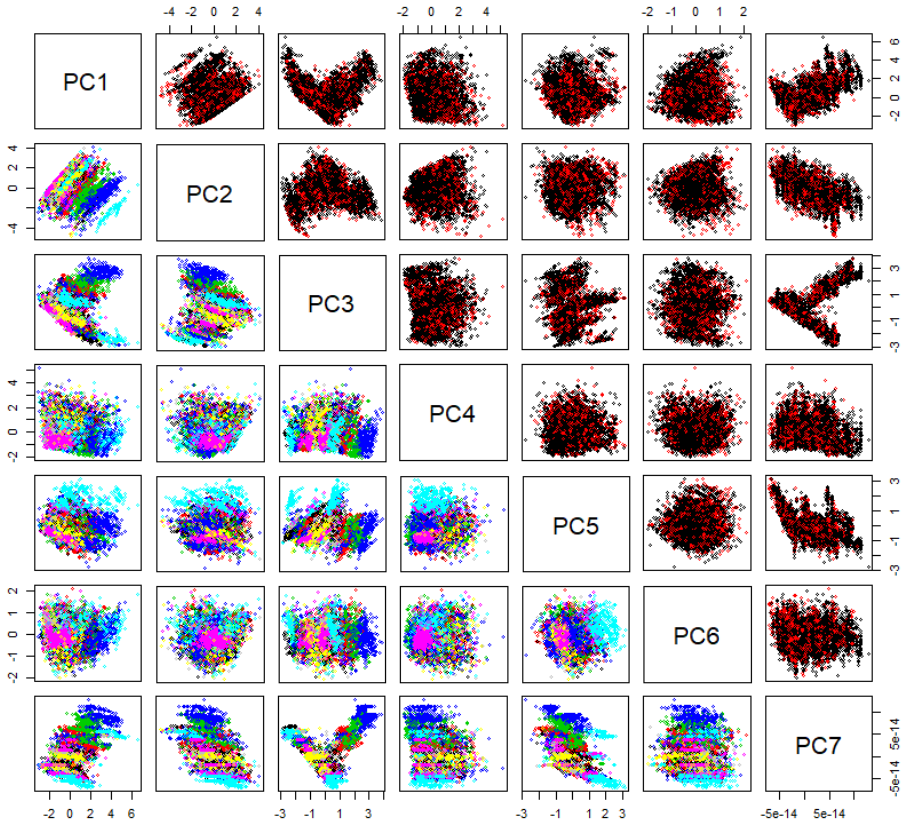


図 4 -14 主成分分析の各成分と低血糖イベント，患者の関係

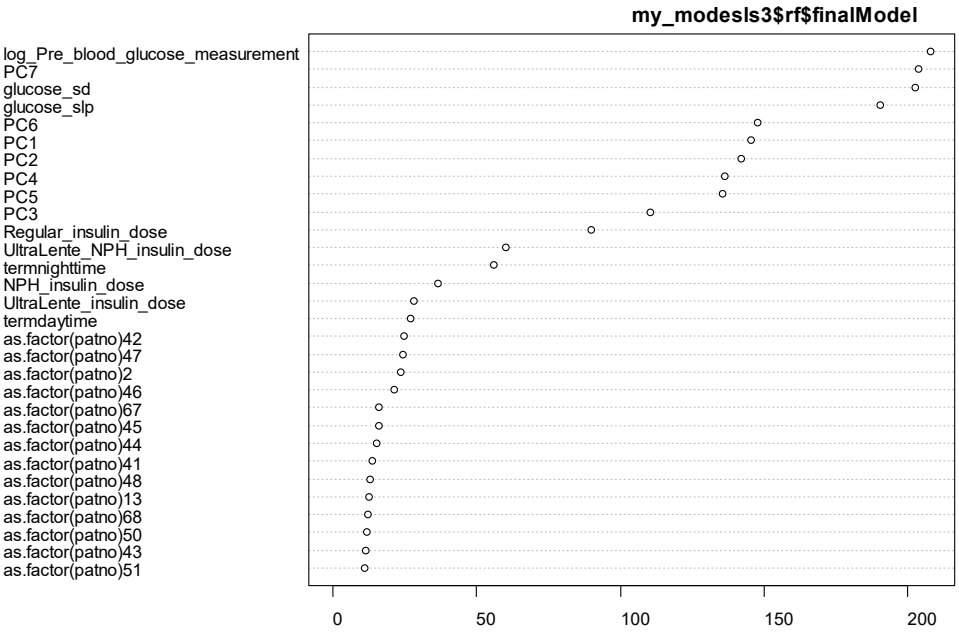


図 4 -15 RandomForest の寄与した変数の視覚化

表 4 -17 主成分分析の結果を用いた学習の要約

	accuracy	recall	specificity	The final values used for the model
knn	0.7676	0.0725	0.9807	k=13
rpart	0.7649	0.1907	0.9409	cp = 0.006993007
lvq	0.7354	0.1341	0.9197	size = 48 and k = 21
svmRadial	0.7654	0	1	sigma(constant) = 0.0006247193 and C = 0.25
gbm	0.7809	0.3694	0.9070	n.trees = 50, interaction.depth = 4, shrinkage(constant) = 0.1 and n.minobsinnode(constant) = 10
rf	1	1	1	mtry = 21
nnet	0.7063	0.2498	0.8462	size = 1 and decay = 1e-04
nb	0.7654	0	1	fL(constant) = 0, usekernel = TRUE and adjust(constant) = 1
xgbTree	0.9812	0.9708	0.9844	nrounds = 250, max_depth = 5, eta = 0.3, gamma(constant) = 0, colsample_bytree = 0.6, min_child_weight(constant) = 1 and subsample = 0.875

表 4 -18 主成分分析の結果を用いた各手法のテストデータの結果

methods	accuracy	recall	specificity	precision	confusion table			
					cel00	cel01	cel10	cel11
knn	0.7611	0.0681	0.9768	0.4375	2476	72	737	56
rpart	0.7591	0.1728	0.9415	0.5801	2341	207	507	286
lvq	0.7336	0.1198	0.9246	0.3033	2270	278	672	121
svmRadial	0.7626	0	1	NaN	2548	0	793	0
gbm	0.7872	0.3619	0.9195	0.5915	2347	201	502	291
rf	0.8207	0.3821	0.9572	0.7627	2459	89	507	286
nnet	0.7052	0.2409	0.8497	0.3284	2276	272	660	133
nb	0.7626	0	1	0.2374	0	2548	0	793
xgbTree	0.7818	0.4792	0.8760	0.5606	2258	290	423	370

D） クラス変数の患者を連続変数の特徴量で代替えする

患者を層に用いる場合，特徴量に対して患者ごとのデータ件数が少ないため，warning が出ている．また，カテゴリである患者（patno）をモデルに含めた場合，学習データに含まれていない新しい患者は patno が欠損となって，予測（識別）ができなくなるため，患者の特徴を他の連続変数で表現したい．そのため，まず，血糖値，インスリン使用量の情報による患者識別を検討した．連続する重複のない 10 回[注32]ごとの空腹時血糖値の平均（glucose_ave），標準偏差（glucose_sd），傾き（glucose_slp），即効型インスリン使用量の平均（Regular_ave），標準偏差（Regular_sd），NPH インスリン使用量の平均（NPH_ave）UltraLente インスリン使用量の平均（UltraLente_ave）の分布は図 4 -16 のとおりである．

注32 実際は，5 回，10 回，20 回について検討を行い，10 回とした．

これを用いて Support Vector Machine により患者の識別を行うと，glucose_sd，glucose_ave，Regular_ave，Regular_sd，NPH_ave，UltraLente_ave を用いた場合の識別結果は図 4 -17 であり，正解率は 0.523 が得られた．これを患者の層の代わりに用いて同様の学習を行ったテスト結果では正解率，再現率共に rf が最良で 0.965，0.355 であり，患者を層とした場合（model1）と同等の結果となった．

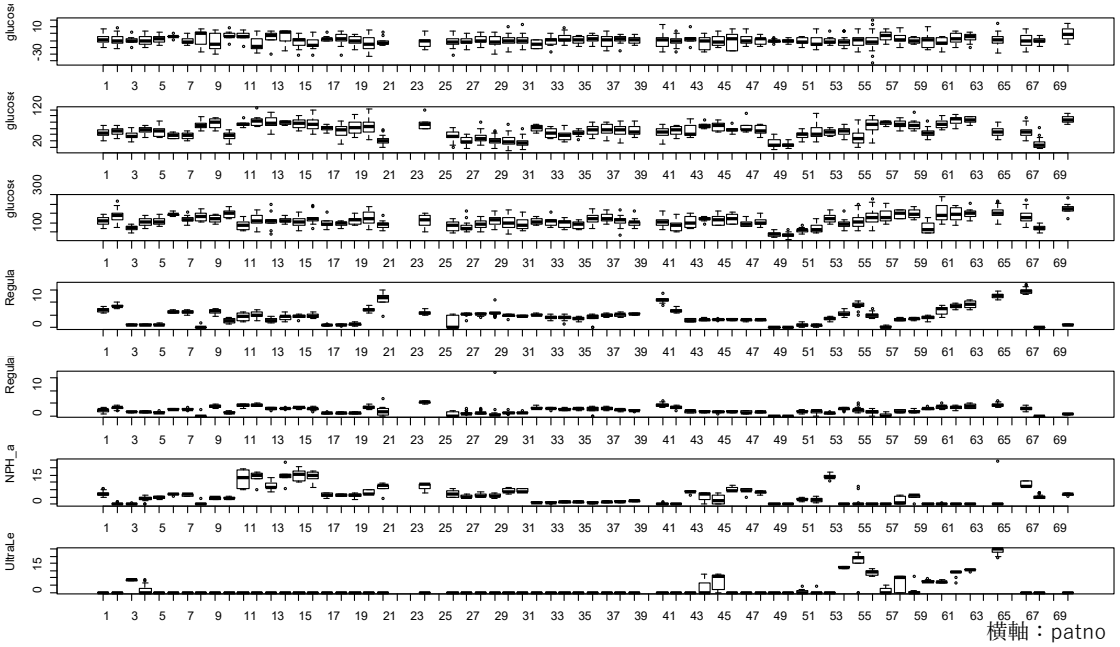


図 4 -16 連続する 10 回ごとの血糖値，インスリン使用量の統計量の分布

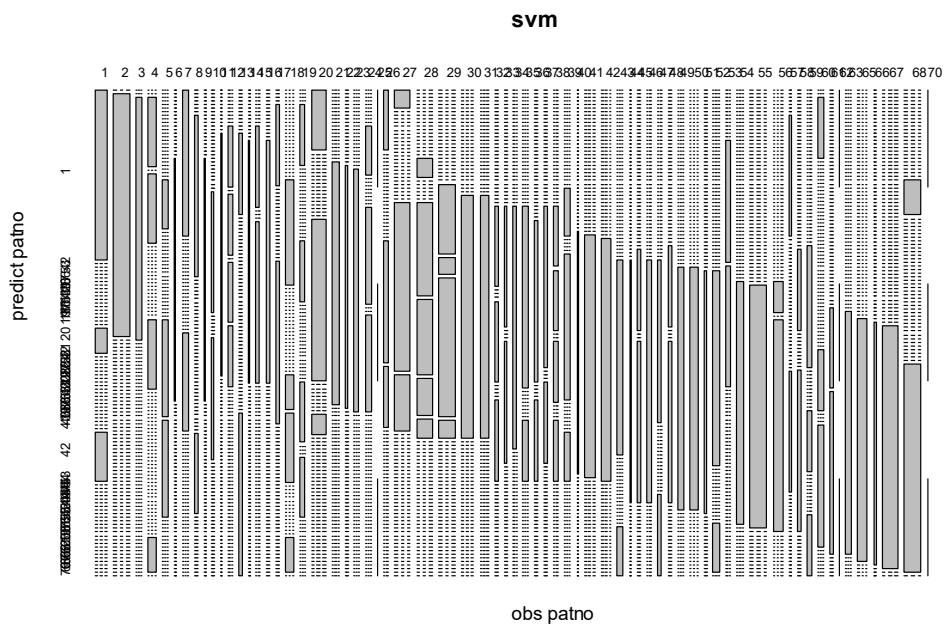


図 4 -17 SVM による患者識別結果

表 4 -19 患者情報を代替えした学習と検証結果の要約

	学習			検証		
	accuracy	recall	specificity	accuracy	recall	specificity
knn	0.8995	0.0478	0.9966	0.9454	0.0182	0.9951
lvq	0.8976	0	1	0.9491	0	1
rpart	0.9209	0.3051	0.9912	0.9602	0.3364	0.9937
svmRadial	0.9251	0.3162	0.9945	0.9621	0.3182	0.9966
gbm	0.9251	0.3309	0.9929	0.9597	0.3182	0.9941
rf	0.9503	0.5221	0.9992	0.9648	0.3545	0.9976
nnet	0.8980	0.0184	0.9983	0.9482	0	0.9990
nb	0.9078	0.1618	0.9929	0.9509	0.1727	0.9927
xgbTree	0.9255	0.3346	0.9929	0.9570	0.3000	0.9922

E） 特徴量の絞り込み

主成分分析による要因などを特徴量に加えたため，特徴量によりユニークになるレコードが多く生じており，学習において警告が出ている．また，追加した特徴量は観察された特徴量から合成したものであるため，相関を生じやすくなっている．まずフィルターメソッド（filter Method）として，学習データ[注 33]について全ての連続変数の組合せの Pearson 相関係数を求めて確認した結果（表 4 -20），UltraLente_insulin_dose，PC7，PC3 は互いに相関が強く，PC1 は複数の特徴量と相関が認められることから，UltraLente_insulin_dose，PC7 及び PC1 のいずれかは除くのが適切と考えられる．

次に，重要度から用いる特徴量を検討するため，ラッパーメソッド（Wrapper Method）として全ての特徴量に 1 回前の空腹時血糖値と regular_insulin 使用量[注 34]を加えたモデル（model3）を用いた rf[注 35]を行い，各特徴量の重要度（学習データ，MeanDecreaseGini）を確認すると（図 4 -18 左），主成分分析による特徴量は全て上位 12 位以内であり， PC7 と PC1 は上位 1 番目と 6 番目で同程度の影響度である．また，インスリンの使用量は Regular_insulin_dose が 13 番目，UltraLente_NPH_insulin_dose が 19 番目，NPH_insulin_dose が 20 番目，UltraLente_insulin_dose が 23 番目であるが，log_Pre_blood_glucose_measurement を 1 とすると Regular_insulin_dose は 0.33，UltraLente_NPH_insulin_dose は 0.22 程度である．Filter Method の結果と合わせて，Regular_insulin_dose，log_Pre_blood_glucose_measurement，glucose_slp，glucose_sd，pre_Pre_blood_glucose_measurement，pre_Regular_insulin_dose，

注33 特徴量の選択には学習データを用いること．全データを用いては（テストデータを含んでは）ならない．

注34 pre_Pre_blood_glucose_measurement 及び pre_Regular_insulin_dose は 1 回前の空腹時血糖値と regular_insulin 使用量

注35 全般的に rf の成績が良く，重要度がデフォルトで出力できるため．

term，PC1，PC2，PC3，PC4，PC5，PC6，pRegular_ave，pglucose_ave，pglucose_sd，pNPH_ave，pUltraLente_ave（model4）を用いることとする．患者に関する特徴量である pRegular_ave，pglucose_ave，pglucose_sd，pNPH_ave，pUltraLente_ave の代わりに患者（patno）を特徴量に用いるのが良さそうであるが，第２項 C）で説明した通り，patno をそのまま用いる場合は，必ず予測する患者のデータを学習データの用いる必要が生じる．

表 4 -20 全ての連続変数の組合せの相関係数（学習データ，相関係数≧0.5）

fact1	fact2	pearson
UltraLente_insulin_dose	PC7	0.9279
PC3	PC7	0.8878
UltraLente_insulin_dose	PC3	0.7958
UltraLente_NPH_insulin_dose	PC1	0.7957
Regular_insulin_dose	PC1	0.7324
Regular_insulin_dose	PC5	0.6639
log_Pre_blood_glucose_measurement	glucose_slp	0.6552
log_Pre_blood_glucose_measurement	PC1	0.6422
NPH_insulin_dose	UltraLente_NPH_insulin_dose	0.6389
log_Pre_blood_glucose_measurement	glucose_ave	0.6151
UltraLente_insulin_dose	UltraLente_NPH_insulin_dose	0.5477
UltraLente_insulin_dose	PC1	0.5168
glucose_ave	PC6	0.5090
NPH_insulin_dose	PC2	0.5029

ここまで検討してきた model1 から model4 の特徴量を用いて，knn(10)，rpart(10)，lvq(5)，,svmRadial(10)，gbm(5)，rf(8)，nnet(10)，nb(10)， xgbTree(5)の 9 種類の手法[注 36]による学習を行い，テストデータによる検証を行った．なお，学習には model3 の検討に用いた学習データを用いた[注 37]．結果（表 4 -21）から，いずれの model でも正解率及び再現率については gbm，rf，xgbtree が比較的良い成績であった．これらは決定木に分類される手法であるが，特徴量の追加による改善は僅かであった．適合率については，svmRadial が最もよく，次いで rf となったが，gbm，xgbTree は 50%台であった．
model3 と model4 について rf で用いられた特徴量の重要度を確認すると（図 4 -18），model4 で用いていない PC7 を除けば，10 番目までは同じ特徴量が用いられることが確認できる．更に，学習の結果（表 4 -22，図 4 -19）を確認すると，rf 及び xgbTree は正解

注36 () 内の数は探索する各パラメータのサンプル数．例えば，gbm の場合 n.trees と interaction.depth について各々5 通りの値の全ての組合せが試される．

注37 患者の特徴量に 10 回ごとのデータを用いているため，当初のデータから除かれている患者があるため．

率，再現率，特異度，適合率のいずれも 0.95 以上であり，過学習を生じている可能性があるが，gbm 及び svmRadial は検証の値に近い．

以上から，用いる特徴量はクラス変数である患者（patno）を除き，SVM による患者の識別性評価にもちいた連続変数を用い[注 38]，Regular_insulin_dose 以外のインスリン使用量を主成分分析による因子に置き換えた model4 を用い，再現率（recall）を優先する場合，手法は決定木の仲間である gbm を用いるのが良さそうである．

表 4 -21 各特徴量（model）による検証結果の概要（赤字は best）

特徴量	metod	knn	rpart	lvq	svmRadial	gbm	rf	nnet	nb	xgbTree
model1	accuracy	0.7641	0.7929	0.7234	0.7626	0.7932	**0.8141**	0.7647	0.2374	0.7725
	recall	0.0164	0.3607	0.1690	0	0.3695	0.3102	0.1211	1	**0.4325**
	specificity	**0.9969**	0.9274	0.8960	1	0.9250	0.9710	0.9651	0	0.8783
	precision	0.6190	0.6072	0.3358	NaN	0.6054	0.7688	0.5189	0.2374	0.5253
model2	accuracy	0.7644	0.7591	0.7414	0.7626	0.7872	**0.8168**	0.7626	0.7626	0.7785
	recall	0.0227	0.1728	0.1110	0	0.3619	0.3644	0.0025	0	**0.4325**
	specificity	**0.9953**	0.9415	0.9376	1	0.9195	0.9576	0.9992	1	0.8862
	precision	0.6000	0.4790	0.3563	NaN	0.5833	0.7280	0.5000	NaN	0.5419
model3	accuracy	0.7533	0.7567	0.7285	0.8138	0.7860	**0.8204**	0.7645	0.7645	0.7905
	recall	0.0899	0.4891	0.1386	0.2773	**0.4801**	0.3723	0	0	0.4390
	specificity	0.9577	0.8391	0.9102	0.9790	0.8802	0.9585	1	1	0.8988
	precision	0.3955	0.4835	0.3224	**0.8030**	0.5524	0.7342	NaN	NaN	0.5719
model4	accuracy	0.7533	0.7766	0.7464	0.7999	0.7851	**0.8213**	0.7645	0.7576	0.7950
	recall	0.0822	0.4326	0.1027	0.1605	**0.4750**	0.3094	0	0.2413	0.4467
	specificity	0.9601	0.8826	0.9446	0.9968	0.8806	0.9790	1	0.9166	0.9023
	precision	0.3879	0.5315	0.3636	**0.9398**	0.5506	0.8197	NaN	0.4712	0.5849

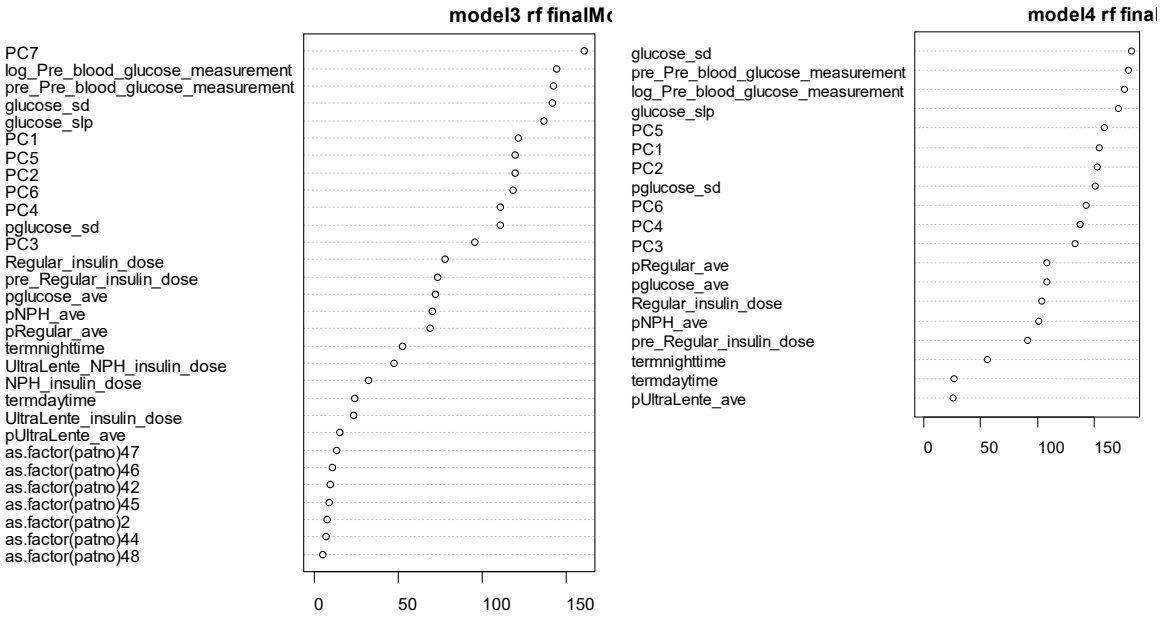


図 4 -18 rf における model3 と model4 の特徴量の重要度比較

注38 カテゴリである患者（patno）をモデルに含めた場合，学習データに含まれていない新しい患者は patno が欠損となって，予測（識別）ができなくなる．また，SVM による各 patno の分類確率を特徴量とすることも考えられるが，ここでは患者識別に有用な観測値を患者の特徴を反映する特徴量として用いることとした．

表 4 -22 model4 の学習結果の要約

method	accuracy	recall	specificity	precision	学習時間（分）	The final values used for the model
knn	0.7723	0.1307	0.9673	0.5489	1.2	k = 13
rpart	0.7782	0.4356	0.8823	0.5295	0.1	cp = 0.005825243
lvq	0.7394	0.0945	0.9355	0.3080	6.3	size = 10, k = 16
svmRadial	0.7989	0.1463	0.9972	0.9417	13.1	sigma = 0.04052906, C = 1
gbm	0.8470	0.6362	0.9111	0.6850	1.8	n.trees = 250, interaction.depth = 5, shrinkage = 0.1, n.minobsinnode = 10
rf	0.9964	0.9845	1	1	12.3	mtry = 2
nnet	0.7669	0	1	NaN	34.7	size = 1, decay = 0.001
nb	0.7540	0.2440	0.9091	0.4493	1.4	fL = 0, usekernel = True, adjust = 1
xgbTree	0.9949	0.9929	0.9955	0.9852	32.7	nrounds = 250, max_depth = 5, eta = 0.3, gamma = 0, colsample_bytree = 0.6, min_child_weight = 1, subsample = 1

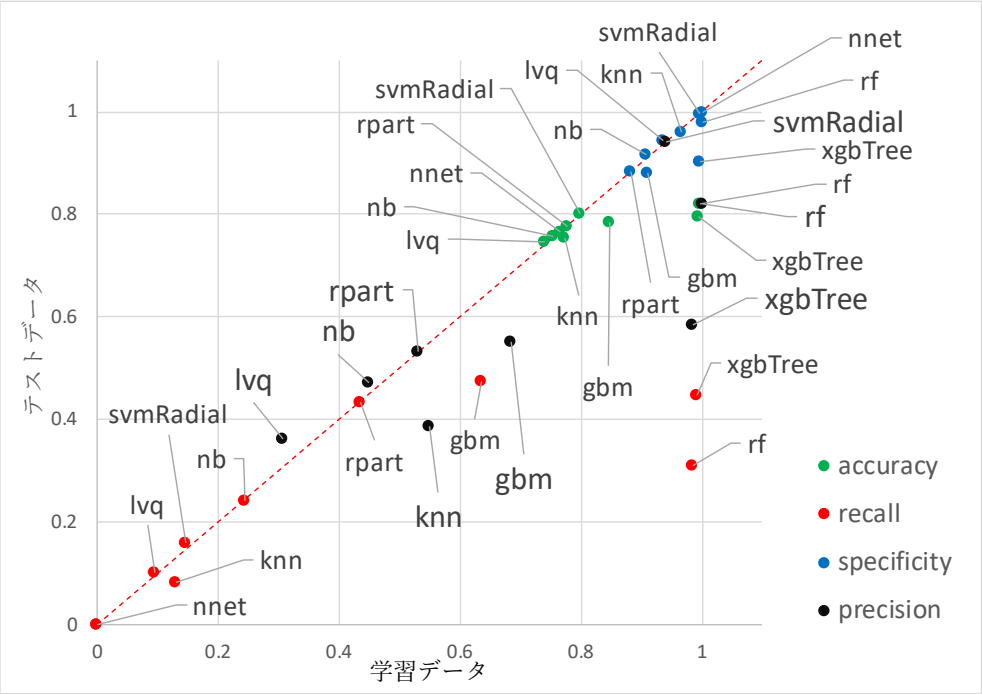


図 4 -19 model4 の学習データと検証データによる各評価指標

第 4 節 手法を選択してチューニングする

先の第 4 章第 3 節第 2 項で概ね用いる特徴量と手法を絞り込むことができたが，全ての手法はパラメータをデフォルト又は自動設定により行っており，本来の性能が得られていない可能性が高い．そこで，gbm に加えて，教師ありの線形分類手法の代表である SVM を扱い，最終的にいずれかを選択することとする．

また，ここまでは R を用いてきたが，機械学習に Python を用いる読者も多いと思われるので，以降は Python 3.6 と Python の機械学習ライブラリである scikit-learn を用いることとする．

第 1 項 手法とパラメータの説明

A） Gradient Boosting Machine（gbm）

gbm については，第 3 節第 1 項 A）で説明している．ここでの学習には scikit-learn 0.22.1 の sklearn.ensemble .GradientBoostingClassifier を用いる．使用できるパラメータを「名称（デフォルト値）：説明」のフォーマットで以下に示す[27] ．

- loss（deviance）：最適化する損失関数．{'deviance'，'exponential'}から選択．
- learning_rate（0.1）：各ツリーの貢献度を learning_rate だけ縮小する．小さくするとツリーの数は増える．n_estimators とトレードオフの関係．
- n_estimators（100）：実行するブースティングステージの数．通常，大きな数を指定するとパフォーマンスが向上する．
- subsample（1.0）：各ステップの決定木の構築に関係するデータの割合．1.0 より小さいと確率的勾配ブースティングになる．
- criterion（friedman_mse）：分割の品質を測定する方法．サポートされる基準は，Friedman による改善スコアを伴う平均二乗誤差の'friedman_mse'，平均二乗誤差の'mse'，及び平均絶対誤差の'mae'．デフォルト値の'friedman_mse'は，適切な近似値を提供できる場合が多いためデフォルト値が用いられることが多い．
- min_samples_split（2）：内部ノードを分割するために必要なサンプルの最小数．整数の場合，最小のサンプル数として扱われ，小数の場合，各分割のサンプルのに対する割合となる．
- min_samples_leaf（1）：葉として必要なサンプルの最小数．左右の枝分かれの葉がこの数より多いようにする
- min_weight_fraction_leaf（0）：リーフノードに存在する必要がある（すべての入力サンプルの）重みの合計の最小重み付き割合．指定されていない場合，サンプルの重みは重量は等しくなる．

- init（None）：初期予測の計算に使用される推定器オブジェクト．estimator または 'zero'を設定する．初期予測を計算するもの．
- random_state（None）：乱数のシード値の指定．比較する場合や再現性が必要な場合は必ず設定する．
- max_features（None）：最適な分割を探す際に考慮する機能．整数を設定した場合，各分割での max_features 機能を検討する．大きいほど過学習が生じやすい．float の場合，int(max_features * n_features)フィーチャが各スプリットで考慮される．auto の時は max_features=n_features，sqrt の時は max_features=sqrt(n_features)，log2 の時は max_features=log2(n_features)，None の時は max_features=n_features
- ccp_alpha（0.0）：最小コスト複雑性プルーニングに使用される複雑性パラメータ．ccp_alpha よりも小さい最大のコスト複雑度を持つサブツリーが選択される． デフォルトではプルーニングは実行されない．
- verbose（0）：詳細出力を有効にする．1 の場合，進行状況とパフォーマンスが時々表示される（ツリーが多いほど，頻度は低くなる）．1 より大きい場合，すべてのツリーの進行状況とパフォーマンスを出力する．
- warm_start（False）：True の場合，前の呼び出しの解を再利用してアンサンブルに合わせて推定器を追加する．それ以外の場合は，前の解を消去する．
- validation_fraction（0.1）：早期停止のための妥当性確認として設定される学習データの割合．0 から 1 の間でなければならない．n_iter_no_change に整数が設定されている場合にのみ使用される．
- n_iter_no_change（None）：検証スコアが改善されない場合にトレーニングを終了するために早期停止を使用するかどうかを決定するために使用される．デフォルトでは，早期停止を無効にするために None に設定されている．数値に設定すると，トレーニングデータの validation_fraction サイズを検証として確保し，以前のすべての n_iter_no_change の反復回数で検証スコアが改善されない場合にトレーニングを終了する．
- tol（1e-4）：損失が n_iter_no_change の反復で少なくとも tol だけ改善されない場合（数値に設定されている場合），トレーニングは停止する．
- min_impurity_decrease（0）：分岐元から分岐先に分かつ際に，あまり impurity が下がらないようならば，その分岐を抑制するためのオプション．0 から 1 の float 値にて，条件分岐をするために最低限必要な impurity 低下を設定する．0（default 値）の場合は抑制なし．

B） Support Vector Machines（SVM）

SVM については，第 3 節第 1 項 A）で説明している．ここでの学習には scikit-learn 0.22.1 のクラス分類を行う sklearn.svm .SVC を用いる．使用できるパラメータを gbm と同様の形式で以下に示す．[28]

- C(1.0)：正則化パラメータ． 正則化の強度は値に反比例する，必ず正の値でなければならない．ペナルティは L2 正則化が用いられる．
- kernel（'rbf'）：アルゴリズムで使用されるカーネルタイプを指定する．'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'，または呼び出し可能のいずれかでなければならない．何も指定されていない場合，「rbf」が使用される．callable が与えられた場合，それはデータ行列からカーネル行列を事前計算するために使用される．その行列は，（n_samples，n_samples）形状の配列でなければならない．
- degree:int（3）：多項式カーネル関数の次数（「poly」）．他のすべてのカーネルでは無視される．
- gamma（scale）：{'scale'，'auto'」}または float．kernel のタイプ 'rbf '，'poly'，'sigmoid'の係数．
    - 「scale」（デフォルト）の場合，ガンマの値として 1 /（n_features * X.var()）が使用される．
    - 'auto'の場合，1 / n_features を使用する．
- coef0（0.0）：カーネル関数の独立した用語．'poly' と 'sigmoid'のみで使われる．
- shrinking（True）：縮小ヒューリスティックを使用するかどうか．
- probability（False）：確率推定を有効にするかどうか．fit を呼び出す前にこれを有効にする必要があり，内部で 5 倍の交差検証を使用し，predict_proba が predict と矛盾する可能性があるため，このメソッドの速度が低下する．
- tol:float（1e-3）：停止基準の許容範囲．
- cache_size：オプション，カーネルキャッシュのサイズ（MB 単位）を指定できる．
- class_weight{dict [注 39], 'balanced'}：オプション，SVC のクラス i のパラメータ C を class_weight [i] * C に設定できる．指定しない場合，すべてのクラスに重み 1 が割り当てられる．「バランス」モードでは，y の値を使用して，n_samples /（n_classes * np.bincount（y））として入力データのクラス頻度に反比例する重みを自動的に調整する．

注39 Python の辞書型のデータ．

- verbose（False）：詳細出力を有効にする．この設定は，libsvm のプロセスごとのランタイム設定を利用する．この設定を有効にすると，マルチスレッドコンテキストで適切に動作しない場合がある．
- max_iter（-1）：ソルバー内の反復のハード制限，制限なしの場合は-1.
- decision_function_shape（ovr）:‘ovo’又は ‘ovr’，他のすべての分類子として形状（n_samples，n_classes）の one-vs-rest（'ovr'）決定関数，または形状（n_samples を持つ libsvm の元の one-vs-one（'ovo'）決定関数を返す，n_classes *（n_classes-1）/ 2）．ただし，1 対 1（'ovo'）は常にマルチクラス戦略として使用される．
- break_ties（False）：設定が true，decision_function_shape = 'ovr'，及びクラス数>2 の場合，predict は decision_function の信頼値に従って関係を破る．そうでない場合，関連付けられたクラスの最初のクラスが返される．単純な予測と比較すると，関係の破壊には比較的高い計算コストがかかることに注意が必要．
- random_state:（None）：オプション，RandomState インスタンスまたは None を設定．確率推定のためにデータをシャッフルするときに使用される擬似乱数ジェネレーターのシード．int の場合，random_state は乱数ジェネレーターによって使用されるシードである．None の場合，乱数ジェネレーターは np.random によって使用される RandomState インスタンスである．パラメータチューニング，手法の比較など再現性が必要な場合は明示的に設定する．

第 2 項 チューニング方法と評価指標

調整が必要なパラメータ数が多いため，チューニングはグリッドサーチによって行う．グリッドサーチは，機械学習モデルのパラメータチューニングを自動で行う手法である．調整するパラメータ，その値の範囲，変化させる量を，評価指標（第 4 章第 3 節第 2 項 A）参照）を設定し，そのすべての組み合わせを計算して最も良いものを選べる．本章では，Python の機械学習ライブラリ scikit-learn の sklearn.model_selection.GridSearchCV 関数を使用する．

GridSearchCV の“CV”は Cross-Validation（交差検証法）である．GridSearchCV は，パラメータを最適化したい scikit-learn の学習モデル名と，チューニングするパラメータとその範囲，交差検証法の分割数などを引数と与えることで，グリッドサーチの汎用的な環境を与える．GridSearchCV で使用できるパラメータは以下がある．「名称（デフォルト値）：説明」のフォーマットで以下に示す[29]

- estimator：必須，チューニングを行うモデル．「モデル名(パラメータ)」の形式で与えることができる．

- param_grid：必須，チューニングするパラメータ候補値を「パラメータ名：候補値リスト」の辞書で与える．
  例 param_grid ={'max_depth': list(range(1, 20)), 'criterion': ['gini', 'entropy'], }
- scoring：string，callable[注40]，list/tuple，dict 又は None．デフォルトは'None'．テストセットの予測を評価するための単一の文字列（モデル評価ルールを参照）または callable（スコアリング戦略を参照）．複数のメトリクスを評価するには，（一意の）文字列のリストか，キーとして名前，値として呼び出し可能オブジェクトを含む辞書を指定する．なしの場合，estimator で指定された学習モデルのデフォルトの方法が使用される．
- n_jobs（None）：整数または None．同時実行数(-1 にすると CPU コア数で同時実行)．
- pre_dispatch :オプション．整数または文字．同時実行数にディスパッチされるジョブの数を設定する．この数を減らすことは，CPU が処理できるよりも多くのジョブがディスパッチされるときにメモリ消費の爆発を避けるのに役立つ．'None'の場合，すべてのジョブがすぐに生成される．整数は，生成されるジョブの正確な総数となる．'2 * n_jobs'のように，n_jobs の関数として式を文字列で与えることができる．
- iid（False）：True の場合，各テストセットのサンプル数で重み付けされた，フォールド全体の平均スコアを返す． この場合，データはフォールド全体に均等に分布していると想定され，最小化される損失はサンプルあたりの総損失であり，フォールド全体の平均損失ではない．deprecated は False と同じ．
- cv（5）：交差検証の分割数（KFold のフォールドの数）．オプションであり，指定しない場合はデフォルトの 5 が用いられる．整数または入力がない場合で，推定器が分類器，y がバイナリまたはマルチクラスの場合は StratiifiedKFold が使用される．そのほかの場合はすべて，KFold が使用される．
- refit（True）：boolean，string 又は callable．データセット全体で見つかった最もよいパラメータを使用して推定量を再構成する．複数のメトリック評価の場合，これはスコアラーを示す文字列である必要がある．これは，最後に推定量を再設定するための最適なパラメータを見つけるために使用される．最適な推定量を選択する際に最大スコア以外の考慮事項がある場合は，cv_results_を指定して，選択した best_index_を返す関数に refit を設定できる．その場合，best_estimator_及び best_parameters_は返された best_index_に従って設定されるが，best_score_属性は使用できない．再調整された推定器は，best_estimator_属性で使用可能になり，この GridSearchCV インスタンスで直接 predict を使用できる．また，複数のメトリック評価の場合，属性

注40 Python の組み込み関数．オブジェクトの引数が呼び出し可能な場合は True を返し，そうでない場合は False を返す．

best_index_，best_score_，及び best_params_は，refit が設定されている場合にのみ使用でき，これらすべてはこの特定のスコアラーによって決定される．

- verbose：整数，ログ出力レベルの冗長性を制御する．大きいほど，より多くのメッセージが表示される（大きいほど出力は長くなる）．0 を指定すると学習の進行度が非表示となる．
- error_score（np.nan）：'raise'または numeric（数値）．エスティメータフィッティングでエラーが発生した場合にスコアに割り当てる値．'raise'に設定すると，エラーが発生する．数値が指定されると，FitFailedWarning が発生する．このパラメータは，常にエラーを発生させる再調整ステップには影響しない．
- return_train_score（False）：False の場合，cv_results_属性にはトレーニングスコアを含まない．トレーニングスコアの計算は，さまざまなパラメータ設定がオーバーフィット/アンダーフィットのトレードオフにどのように影響するかについての洞察を得るために使用される．ただし，トレーニングセットのスコアの計算は計算コストが高くなる可能性があり，最高の汎化パフォーマンスをもたらすパラメータを選択するために厳密には必要でない．

グリッドサーチは与えられたパラメータの全ての組み合わせで学習を行うため，計算コストが高い（時間を要する）ので，パラメータの組み合わせ数はあまり多くはできない．また，探索するパラメータは例えば rf の場合，作成する決定木の深さを 1，2，3，4，5 のように与えて 3 の場合に最もよい性能が得られる場合はそのままでよいが，5 の場合であれば，試していない 5 より大きな値で良い性能が得られるかもしれない．また，作成する決定木の数を 20, 100, 1000 のように与えて性能が 100>1000>20 である場合は，100 から 1000 の間により良いパラメータがあるかもしれないが，パラメータによる精度などの評価指標の変化量が小さければそのまま 100 を用いてもよい．このため，通常，最初は 1 パラメータにつき 3～5 個の値を最大と最小の間は広く設定して性能が得られるパラメータの見当をつけて状況に応じて絞り込む．また，パラメータ間で相互作用があることが少なくないため，複数のパラメータを調整する場合は，各組み合わせを可視化して確認するとよい．例えば図 4-21 では，gbm のパラメータ n_eatimators，max_depth と learning rate について，精度，再現率，適合率の値の変化を可視化している．gbm の評価指標に用いている精度を見ると，n_eatimators の値によらず，max_depth は 3，learning rate は 0.05～0.5 が最良となっており，n_eatimators は 1000 と 2000 でほとんど同じであることが確認できる．

第 3 項 パラメータのチューニングを行う

下記の手順によりチューニングを行った．以下に手順ごとのコード及び性能を示す．なお，gbm では手順 2 で 6 パターン実施したため，6 パターンの結果を表示する．SVM では手順

2 のスケール変換で 3 つの方法を実施したが，手順 3 において変換方法を選んでいる．なお，チューニング中はテストデータによるチューニング結果の確認は行わない．チューニングにより選択した結果のパラメータ設定，学習器の確認のみテストデータを用いている．

モデル・特徴量データ：
Model4 の特徴量 Regular_insulin_dose，log_Pre_blood_glucose_measurement，glucose_slp，glucose_sd，pre_Pre_blood_glucose_measurement，pre_Regular_insulin_dose，term，PC1，PC2，PC3，PC4，PC5，PC6，pRegular_ave，pglucose_ave，pglucose_sd，pNPH_ave，pUltraLente_ave を用いる．term は morning，daytime，nighttime の 3 つのカテゴリからなるクラス変数であるので，one-hot encoding [注 41]により各々0,1 の値を持つ 3 つの変数とする．その結果，用いる特徴量は
Regular_insulin_dose，log_Pre_blood_glucose_measurement，glucose_slp，glucose_sd，pre_Pre_blood_glucose_measurement，pre_Regular_insulin_dose，morning，daytime，nighttime，PC1，PC2，PC3，PC4，PC5，PC6，pRegular_ave，pglucose_ave，pglucose_sd，pNPH_ave，pUltraLente_ave となる．

A）【Gradient Boosting Machine（gbm)】

勾配ブースティングマシーン（gbm）を用いた学習とチューニングを行う．

- 手順 1）確認のため，全てデフォルトのパラメータを用いて学習を行う[注 42]． 精度として 0.820 が得られた．
- 手順 2）低血糖ありが少ない不均衡データであるため，Gridsearch によるパラメータチューニングでは，低血糖なしの正解が多く，ありの正解が少なくなるパラメータが選ばれやすい．不均衡への対処として，低血糖ありを 2，なしに 1 の重みづけを行う．
- 手順 3）弱学習器である決定木の数となるパラメータ「n_estimators」について，1，100（デフォルト），500，1000，1500，2000，3000，4000，10000 と設定を変えて結果を確認する．結果は図 4 -20 に示す．学習データの精度（accuracy）は n_estimators の値が大きくなるにしたがって大きくなり，4000 で 1 となった．accuracy=1 は過学習が生じていると考えられ，accuracy の変化量から n_estimators が 500，1000，2000 について次のパラメータチューニングを行う．

注41 第 2 章参照.

注42 第 4 章第 3 節第 2 項 E）のパラメータを用いてもよい．有効なパラメータ及びデフォルト値はライブラリのバージョンによって変わることがあるため，デフォルトであってもパラメータとして設定している．

- 手順 4）n_estimators が 500，1000，2000 について，各決定木の影響度となる learning_rate（デフォルト 0.1），各決定木の深さの max_depth（デフォルト 3）を，GridSearchCV（sklearn.model_selection.GridSearchCV）を用いてハイパーパラメータチューニングを行う．なお，過学習の傾向が強いため，min_samples_leaf=2（デフォルト 1），min_samples_split=3（デフォルト 2）とする．探索するパラメータの候補は第 3 節第 2 項 E）を参考にして，learning_rate を 0.01，0.05, 0.1, 0.5, 1，2，3，max_depth を 1，2，3，4，5 とする．交差検証の分割数はデータ量が多いとは言えないためデフォルトのまま，cv=5 とする．詳細は表 4 -23 に示す．なお，gbm のモデル設定は青， GridSearchCV のパラメータのデフォルトは灰色，チューニングパラメータは赤で示している．
- 出力が長くなるので，learning_rate は 0.05，0.1，0.5，1，max_depth は 2，3，4 についてチューニングの結果を表 4 -24 に示す．GridSearchCV では，n_estimators が 500，1000 及び 2000 いずれも，learning_rate は 0.01，max_depth は 2 が選ばれている．cross validation のデフォルトの評価基準の精度は，イベントが少ない不均衡データでは specificity の影響が強く，再現率が悪くなる．実装では，低血糖イベントの再現率（感度）を重視するため，テストデータを用いてパラメータの組み合わせの結果を図 4 -21 により視覚的に評価する．再現率と適合率のバランスから，n_estimators=500，learning_rate=0.05，max_depth=4 を選択する．
- 手順 5）手順 4 では視覚的な選択を行ったが，モデルの評価指標に再現率（感度）を用いることで，自動化が可能である．パラメータ「scoring='recall'」[注 43]を追加して，手順 4 と同じ GridSearchCV を行うと（結果は表 4 -24 参照），n_estimators=500 の Best parameters は，learning_rate: 0.05，max_depth: 4 となった．図 4 -21 にチューニングの評価指標に精度を用いた場合に選ばれたパラメータを黄色，再現率を用いた場合を黄緑色で示している．また，特徴量の影響度を図 4 -22 により確認する．実装では Regular insulin の使用量を調整するモデルとして使用するため，Regular_insulin_dose の影響度がある程度大きいことが期待されるが，19 の特徴量中 7 番目（6.5%）となっている．
- 手順 6）テストデータによる確認を行う．手順 4 で選択したパラメータと手順 5 で自動的に選ばれたパラメータは同じであるので，手順 5 から best_estimator_プロパティを用いることでチューニングされたモデルを利用できる．その結果を表 4 -25 に示す．再現率は，学習時の 0.765 から 0.490 と約 1/3 低下しており，過学習が生じていたことがうかがわれる．

注43 sklearn の scoring オプションに用いる recall は，分類クラスが 0,1 で与えられている場合，デフォルトで 1 の場合の再現率（recall）が用いられる．

表 4 -23 学習モデルのコードとパラメータ（デフォルトは灰色，探索的に評価する設定は赤）

手法	手順	コード（パラメータ）
gbm	手順 1	gbm=GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None, learning_rate=0.1, loss='deviance', max_depth=3, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=100, n_iter_no_change=None, random_state=1010, subsample=1.0, tol=0.0001, validation_fraction=0.1, verbose=0, warm_start=False)
	手順 2	gbm.fit=(X_train, y_train, sample_weight = sample_weights)
	手順 3	gbm=GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None, learning_rate=0.1, loss='deviance', max_depth=3, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=[1,500,1000,1500,2000,3000,4000,10000],n_iter_no_change=None, random_state=1010, subsample=1.0, tol=0.0001, validation_fraction=0.1, verbose=0, warm_start=False) gbm.fit=(X_train, y_train, sample_weight = sample_weights)
	手順 4	gbmgs=GridSearchCV(cv=5, error_score=nan, estimator=GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None, learning_rate=0.1, loss='deviance', max_depth=3, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=2, min_samples_split=3, min_weight_fraction_leaf=0.0, n_estimators=[500,1000,2000], n_iter_no_change=None, presort='deprecated', random_state=1010, subsample=1.0, tol=0.0001, validation_fraction=0.1, verbose=0,warm_start=False), iid='deprecated', n_jobs=None,param_grid={'learning_rate': [0.01, 0.05, 0.1, 0.3, 0.5, 1, 2, 3], 'max_depth': [1, 2, 3, 4, 5]},pre_dispatch='2*n_jobs', refit=True, return_train_score=False, scoring=None, verbose=0) gbmgs.fit=(X_train, y_train, sample_weight = sample_weights)
	手順 5	gbmgs=GridSearchCV(cv=5, error_score=nan, estimator=GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None, learning_rate=0.1, loss='deviance', max_depth=3, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=2, min_samples_split=3, min_weight_fraction_leaf=0.0, n_estimators=[500,1000,2000], n_iter_no_change=None, presort='deprecated', random_state=1010, subsample=1.0, tol=0.0001, validation_fraction=0.1, verbose=0, warm_start=False), iid='deprecated', n_jobs=None, param_grid={'learning_rate': [0.01,0.05, 0.1, 0.3,0.5, 1,2,3], 'max_depth': [1, 2, 3, 4, 5]}, pre_dispatch='2*n_jobs', refit=True, return_train_score=False, scoring='recall', verbose=0) gbmgs.fit=(X_train, y_train, sample_weight = sample_weights)
SVM	手順 1	SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=1010, shrinking=True, tol=0.001, verbose=False)
	手順 2	同上

	手順 3	SVC(C=1.0, break_ties=False, cache_size=200, class_weight='balanced', coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=1010, shrinking=True, tol=0.001, verbose=False)
	手順 4	GridSearchCV(cv=5, error_score=nan, estimator=SVC(C=1.0, break_ties=False, cache_size=200, class_weight='balanced', coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=1010, shrinking=True, tol=0.001, verbose=False), iid='deprecated', n_jobs=None, param_grid={'C': [0.1, 1, 10, 100], 'gamma': [0.001, 0.01, 0.1, 1]}, pre_dispatch='2*n_jobs', refit=True, return_train_score=False, scoring=None, verbose=0)
	手順 5	GridSearchCV(cv=5, error_score=nan, estimator=SVC(C=1.0, break_ties=False, cache_size=200, class_weight='balanced', coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=1010, shrinking=True, tol=0.001, verbose=False), iid='deprecated', n_jobs=None, param_grid={'C': [0.1, 1, 10, 100], 'gamma': [0.001, 0.01, 0.1, 1]}, pre_dispatch='2*n_jobs', refit=True, return_train_score=False, scoring='recall', verbose=0)

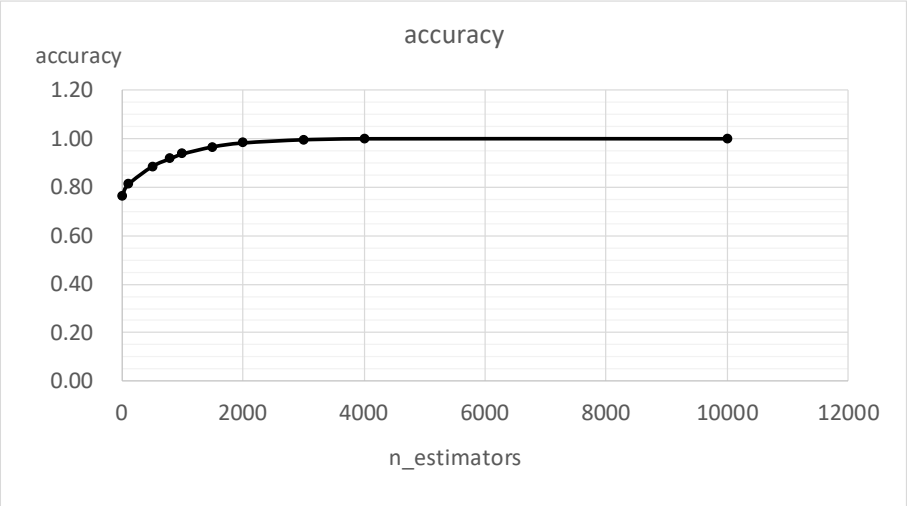


図 4 -20 gbm の n_estimators の検討結果

表 4 -24 gbm の学習モデルのパラメータ最適化の結果[注 44]

n_estimators /scoring	CV results（スコアの降順でソート）	Tuning results （学習データ）
500 /None	0.585(+/-0.321)for{'learning_rate':0.01,'max_depth':2} 0.564(+/-0.296)for{'learning_rate':0.01,'max_depth':3} 0.524(+/-0.255)for{'learning_rate':0.01,'max_depth':4} 0.534(+/-0.270)for{'learning_rate':0.05,'max_depth':2} 0.495(+/-0.311)for{'learning_rate':0.05,'max_depth':3} 0.479(+/-0.307)for{'learning_rate':0.05,'max_depth':4} 0.501(+/-0.300)for{'learning_rate':0.1,'max_depth':2} 0.481(+/-0.327)for{'learning_rate':0.1,'max_depth':3} 0.486(+/-0.331)for{'learning_rate':0.1,'max_depth':4} 0.508(+/-0.318)for{'learning_rate':0.5,'max_depth':2} 0.504(+/-0.334)for{'learning_rate':0.5,'max_depth':3} 0.504(+/-0.327)for{'learning_rate':0.5,'max_depth':4} 0.502(+/-0.285)for{'learning_rate':1,'max_depth':2} 0.511(+/-0.290)for{'learning_rate':1,'max_depth':3} 0.518(+/-0.292)for{'learning_rate':1,'max_depth':4}	[Best score]: 0.58503 [Best parameters]: {'learning_rate': 0.01, 'max_depth': 2} 正解率: 0.781 適合率: 0.930 再現率: 0.068 混同マトリクス: [[5081 8] [1446 106]]
1000 /None	0.553(+/-0.318)for{'learning_rate':0.01,'max_depth':2} 0.514(+/-0.262)for{'learning_rate':0.01,'max_depth':3} 0.487(+/-0.289)for{'learning_rate':0.01,'max_depth':4} 0.501(+/-0.291)for{'learning_rate':0.05,'max_depth':2} 0.488(+/-0.325)for{'learning_rate':0.05,'max_depth':3} 0.482(+/-0.326)for{'learning_rate':0.05,'max_depth':4} 0.497(+/-0.327)for{'learning_rate':0.1,'max_depth':2} 0.486(+/-0.332)for{'learning_rate':0.1,'max_depth':3} 0.495(+/-0.329)for{'learning_rate':0.1,'max_depth':4} 0.509(+/-0.308)for{'learning_rate':0.5,'max_depth':2} 0.507(+/-0.329)for{'learning_rate':0.5,'max_depth':3} 0.510(+/-0.326)for{'learning_rate':0.5,'max_depth':4} 0.492(+/-0.311)for{'learning_rate':1,'max_depth':2} 0.514(+/-0.292)for{'learning_rate':1,'max_depth':3} 0.521(+/-0.291)for{'learning_rate':1,'max_depth':4}	[Best score]: 0.55340 [Best parameters]: {'learning_rate': 0.01, 'max_depth': 2} 正解率: 0.795 適合率: 0.812 再現率: 0.159 混同マトリクス: [[5032 57] [1306 246]]
2000 /None	0.547(+/-0.283)for{'learning_rate':0.01,'max_depth':2} 0.507(+/-0.283)for{'learning_rate':0.01,'max_depth':3} 0.481(+/-0.298)for{'learning_rate':0.01,'max_depth':4} 0.489(+/-0.325)for{'learning_rate':0.05,'max_depth':2} 0.480(+/-0.340)for{'learning_rate':0.05,'max_depth':3} 0.490(+/-0.341)for{'learning_rate':0.05,'max_depth':4} 0.495(+/-0.336)for{'learning_rate':0.1,'max_depth':2} 0.491(+/-0.331)for{'learning_rate':0.1,'max_depth':3} 0.502(+/-0.328)for{'learning_rate':0.1,'max_depth':4} 0.509(+/-0.305)for{'learning_rate':0.5,'max_depth':2} 0.513(+/-0.314)for{'learning_rate':0.5,'max_depth':3} 0.508(+/-0.331)for{'learning_rate':0.5,'max_depth':4} 0.505(+/-0.297)for{'learning_rate':1,'max_depth':2} 0.517(+/-0.289)for{'learning_rate':1,'max_depth':3} 0.521(+/-0.291)for{'learning_rate':1,'max_depth':4}	[Best score]: 0.54738 [Best parameters]: {'learning_rate': 0.01, 'max_depth': 2} 正解率: 0.808 適合率: 0.836 再現率: 0.223 混同マトリクス: [[5021 68] [1206 346]]

注44 出力が長くなるため，learning_rate を 0.05，0.1，0.5，1，max_depth を 2，3，4 としている．

n_estimators /scoring	CV results（スコアの降順でソート）	Tuning results （学習データ）
500 /Recall	0.180(+/-0.213)for{'learning_rate':0.01,'max_depth':2} 0.218(+/-0.258)for{'learning_rate':0.01,'max_depth':3} 0.255(+/-0.242)for{'learning_rate':0.01,'max_depth':4} 0.278(+/-0.244)for{'learning_rate':0.05,'max_depth':2} 0.274(+/-0.269)for{'learning_rate':0.05,'max_depth':3} 0.287(+/-0.267)for{'learning_rate':0.05,'max_depth':4} 0.278(+/-0.241)for{'learning_rate':0.1,'max_depth':2} 0.265(+/-0.247)for{'learning_rate':0.1,'max_depth':3} 0.265(+/-0.266)for{'learning_rate':0.1,'max_depth':4} 0.260(+/-0.285)for{'learning_rate':0.5,'max_depth':2} 0.238(+/-0.267)for{'learning_rate':0.5,'max_depth':3} 0.231(+/-0.223)for{'learning_rate':0.5,'max_depth':4} 0.263(+/-0.267)for{'learning_rate':1,'max_depth':2} 0.265(+/-0.195)for{'learning_rate':1,'max_depth':3} 0.236(+/-0.212)for{'learning_rate':1,'max_depth':4}	[Best score]: 0.28672 [Best parameters]: {'learning_rate': 0.05, 'max_depth': 4} 正解率: 0.888 適合率: 0.950 再現率: 0.549 混同マトリクス: [[5044 45] [700 852]]
1000 /Recall	0.238(+/-0.258)for{'learning_rate':0.01,'max_depth':2} 0.268(+/-0.234)for{'learning_rate':0.01,'max_depth':3} 0.270(+/-0.234)for{'learning_rate':0.01,'max_depth':4} 0.285(+/-0.227)for{'learning_rate':0.05,'max_depth':2} 0.271(+/-0.259)for{'learning_rate':0.05,'max_depth':3} 0.270(+/-0.256)for{'learning_rate':0.05,'max_depth':4} 0.277(+/-0.272)for{'learning_rate':0.1,'max_depth':2} 0.255(+/-0.243)for{'learning_rate':0.1,'max_depth':3} 0.244(+/-0.237)for{'learning_rate':0.1,'max_depth':4} 0.253(+/-0.293)for{'learning_rate':0.5,'max_depth':2} 0.222(+/-0.261)for{'learning_rate':0.5,'max_depth':3} 0.217(+/-0.197)for{'learning_rate':0.5,'max_depth':4} 0.275(+/-0.265)for{'learning_rate':1,'max_depth':2} 0.267(+/-0.216)for{'learning_rate':1,'max_depth':3} 0.231(+/-0.216)for{'learning_rate':1,'max_depth':4}	[Best score]: 0.28477 [Best parameters]: {'learning_rate': 0.05, 'max_depth': 2} 正解率: 0.832 適合率: 0.845 再現率: 0.345 混同マトリクス: [[4991 98] [1016 536]]
2000 /Recall	0.274(+/-0.247)for{'learning_rate':0.01,'max_depth':2} 0.279(+/-0.240)for{'learning_rate':0.01,'max_depth':3} 0.264(+/-0.266)for{'learning_rate':0.01,'max_depth':4} 0.280(+/-0.255)for{'learning_rate':0.05,'max_depth':2} 0.260(+/-0.248)for{'learning_rate':0.05,'max_depth':3} 0.250(+/-0.247)for{'learning_rate':0.05,'max_depth':4} 0.265(+/-0.286)for{'learning_rate':0.1,'max_depth':2} 0.242(+/-0.236)for{'learning_rate':0.1,'max_depth':3} 0.233(+/-0.214)for{'learning_rate':0.1,'max_depth':4} 0.262(+/-0.292)for{'learning_rate':0.5,'max_depth':2} 0.229(+/-0.245)for{'learning_rate':0.5,'max_depth':3} 0.214(+/-0.197)for{'learning_rate':0.5,'max_depth':4} 0.280(+/-0.267)for{'learning_rate':1,'max_depth':2} 0.275(+/-0.239)for{'learning_rate':1,'max_depth':3} 0.231(+/-0.216)for{'learning_rate':1,'max_depth':4}	[Best score]: 0.28028 [Best parameters]: {'learning_rate': 0.05, 'max_depth': 2} 正解率: 0.856 適合率: 0.878 再現率: 0.447 混同マトリクス: [[4993 96] [859 693]]

		n_eatimators=500						
	max	learning rate						
	depth	0.01	0.05	0.1	0.5	1	2	3
精度	1	0.772	0.761	0.762	0.778	0.788	0.234	0.766
	2	0.779	0.796	0.822	0.902	0.942	0.696	0.416
	3	0.800	0.847	0.889	0.992	1.000	0.649	0.427
	4	0.830	0.906	0.956	1.000	1.000	0.700	0.415
	5	0.860	0.954	0.991	1.000	1.000	0.535	0.499
再現率	1	0.084	0.330	0.430	0.536	0.584	1.000	0.000
	2	0.308	0.537	0.606	0.802	0.911	0.356	0.613
	3	0.430	0.637	0.737	0.985	1.000	0.235	0.617
	4	0.541	0.764	0.894	1.000	1.000	0.386	0.798
	5	0.624	0.885	0.978	1.000	1.000	0.594	0.568
適合率	1	0.585	0.484	0.490	0.525	0.542	0.234	0.000
	2	0.549	0.568	0.623	0.784	0.850	0.351	0.225
	3	0.601	0.685	0.776	0.982	0.999	0.242	0.230
	4	0.668	0.820	0.914	1.000	1.000	0.366	0.257
	5	0.736	0.914	0.982	1.000	1.000	0.273	0.249

		n_eatimators=1000						
	max	learning rate						
	depth	0.01	0.05	0.1	0.5	1	2	3
精度	1	0.772	0.763	0.767	0.783	0.801	0.234	0.766
	2	0.784	0.819	0.850	0.950	0.988	0.696	0.416
	3	0.814	0.888	0.940	1.000	1.000	0.649	0.427
	4	0.851	0.957	0.992	1.000	1.000	0.700	0.415
	5	0.892	0.991	1.000	1.000	1.000	0.535	0.499
再現率	1	0.173	0.427	0.466	0.564	0.612	1.000	0.000
	2	0.451	0.601	0.671	0.908	0.986	0.356	0.613
	3	0.535	0.729	0.863	1.000	1.000	0.235	0.617
	4	0.628	0.897	0.983	1.000	1.000	0.386	0.798
	5	0.718	0.977	1.000	1.000	1.000	0.594	0.568
適合率	1	0.538	0.492	0.501	0.534	0.568	0.234	0.000
	2	0.546	0.615	0.681	0.882	0.964	0.351	0.225
	3	0.619	0.778	0.878	1.000	1.000	0.242	0.230
	4	0.704	0.917	0.982	1.000	1.000	0.366	0.257
	5	0.801	0.986	1.000	1.000	1.000	0.273	0.249

		n_eatimators=2000						
	max	learning rate						
	depth	0.01	0.05	0.1	0.5	1	2	3
精度	1	0.764	0.767	0.772	0.792	0.811	0.234	0.766
	2	0.793	0.848	0.890	0.988	1.000	0.696	0.416
	3	0.838	0.937	0.981	1.000	1.000	0.649	0.427
	4	0.891	0.992	1.000	1.000	1.000	0.700	0.415
	5	0.934	1.000	1.000	1.000	1.000	0.535	0.499
再現率	1	0.287	0.465	0.510	0.593	0.631	1.000	0.000
	2	0.508	0.664	0.760	0.983	1.000	0.356	0.613
	3	0.613	0.854	0.961	1.000	1.000	0.235	0.617
	4	0.726	0.983	1.000	1.000	1.000	0.386	0.798
	5	0.838	1.000	1.000	1.000	1.000	0.594	0.568
適合率	1	0.490	0.502	0.512	0.551	0.589	0.234	0.000
	2	0.564	0.679	0.769	0.966	0.999	0.351	0.225
	3	0.666	0.872	0.959	1.000	1.000	0.242	0.230
	4	0.789	0.983	1.000	1.000	1.000	0.366	0.257
	5	0.874	1.000	1.000	1.000	1.000	0.273	0.249

図 4 -21 Gridsearch の結果（学習データ）

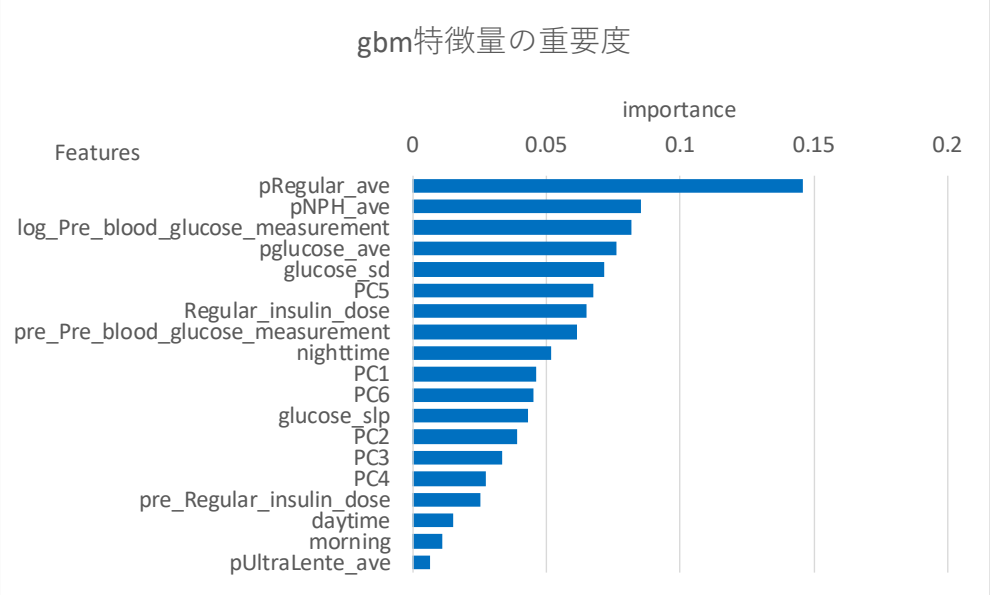


図 4 -22 GBM 手順 5 のモデルでの特徴量の重要度

表 4 -25 gbm のチューニング結果のテストデータによる確認

パラメータ・統計量	値	コード・パラメータ
n_estimators	500	GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None, learning_rate=0.05, loss='deviance', max_depth=4, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=2, min_samples_split=3, min_weight_fraction_leaf=0.0, n_estimators=500, n_iter_no_change=None, presort='deprecated', random_state=1010, subsample=1.0, tol=0.0001, validation_fraction=0.1, verbose=0, warm_start=False)
learning_rate	0.05	
max_depth	4	
sample_weight 使用項目	case_weight	
正解率（テストデータ）	0.800	
適合率（テストデータ）	0.590	
再現率（テストデータ）	0.490	
混同マトリクス	[[2242 263] [394 379]]	

B） 【Support Vector Machines（SVM）】

線形手法であるサポートベクターマシーン（SVM）を用いた学習とチューニングを行う．決定木を用いる gbm との比較のためであるので，出力などを簡略にしている．

- 手順 1）確認のため，全てデフォルトのパラメータを用いて学習を行う．
- 手順 2）SVM は線形モデルであるため，連続変数の特徴量のスケール変換として次の 3 種類を行い，確認として手順 1 のパラメータを用いて学習を行う．
  - A）平均 0，分散 1 となる標準化を行う（sklearn.preprocessing.StandardScaler）
  - B）25%点，50%点，75%点を用いた外れ値に頑健なロバスト標準化を行う（sklearn.preprocessing.RobustScaler）
  - C）最大 1，最小 0 となる正規化を行う（sklearn.preprocessing.MinMaxScaler）
- 手順 3）低血糖ありが少ない不均衡データであるので，手順 2 の結果，学習精度が低く，標準化の性能が評価できない．よって，各々について，class_weight="balanced" を指定してモデルを作成する．（以上，表 4 -26 参照）
- 手順 4）手順 3 の結果，A の「標準化」を選択し，更にパラメータ C，Gamma について，GridSearchCV でハイパーパラメータチューニング（sklearn.model_selection.GridSearchCV）を行う．
- 手順 5）再現率による精度評価を行うためパラメータ「scoring='recall'」を追加して，手順 4 と同じ GridSearchCV を行う．（以上，表 4 -27 参照）

表 4 -26 SVM の結果

処理内容		処理No	Tuning results（学習データ）
「Term」を「morning」，「daytime」，「nighttime」に分けて，それぞれ 0,1 の値を持つ変数とする		1	正解率: 0.767，適合率: 0.692，再現率: 0.006 混同マトリクス:[[5085　　4] [1543　　9]]
標準化を行う（sklearn.preprocessing.StandardScaler）		2-A	正解率: 0.800，適合率: 0.937，再現率: 0.153 混同マトリクス:[[5073　16] [1315　237]]
ロバスト標準化を行う（sklearn.preprocessing.RobustScaler）		2-B	正解率: 0.769，適合率: 0.955，再現率: 0.014 混同マトリクス:[[5088　　1] [1531　21]]
正則化を行う（sklearn.preprocessing.MinMaxScaler）		2-C	正解率: 0.791，適合率: 0.906，再現率: 0.118 混同マトリクス:[[5070　19] [1369　183]]
パラメータ「class_weight="balanced"」を指定してモデル作成	2-A に対して	3-A	正解率: 0.719，適合率: 0.440，再現率: 0.742 混同マトリクス:[[3623 1466] [401 1151]]
	2-B に対して	3-B	正解率: 0.650，適合率: 0.366，再現率: 0.678 混同マトリクス:[[3263 1826] [500 1052]]
	2-C に対して	3-C	正解率: 0.699，適合率: 0.411，再現率: 0.662 混同マトリクス:[[3615 1474] [524 1028]]

表 4 -27 Support Vector Machines の gridsearch の学習コード，パラメータと結果

処理内容	処理No	モデル/結果	テストデータによる結果
3- A に対して，c，gamma，tol について，GridSearchCV でハイパーパラメータチューニング (sklearn.model_selection.GridSearchCV)	4	GridSearchCV(cv=5, error_score=nan, estimator=SVC(C=1.0, break_ties=False, cache_size=200, class_weight='balanced', coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=1010, shrinking=True, tol=0.001, verbose=False), iid='deprecated', n_jobs=None, param_grid= {'C': [0.001, 0.01, 0.1, 1, 10, 100,1000], 'gamma' : [0.0001,0.001, 0.01, 0.1, 1, 10], 'tol':[0.0001, 0.001, 0.01]}, pre_dispatch='2*n_jobs', refit=True, return_train_score=False, scoring=None, verbose=0) **********結果********** 正解率: 1.000 適合率: 0.999 再現率: 1.000 混同マトリクス:[[5088 1] [0 1552]] [Best score]: 0.7655472907431079 [Best parameters]: {'C': 10, 'gamma': 10, 'tol': 0.01} [Best estimator]: SVC(C=10, break_ties=False, cache_size=200, class_weight='balanced', coef0=0.0, decision_function_shape='ovr', degree=3, gamma=10, kernel='rbf',max_iter=-1, probability=False, random_state=1010, shrinking=True, tol=0.01, verbose=False) [CV results]（上位 20 件） 0.766(+/-0.004)for{'C':10,'gamma':10,'tol':0.01} 0.766(+/-0.004)for{'C':100,'gamma':10,'tol':0.01} 0.766(+/-0.004)for{'C':1000,'gamma':10,'tol':0.01} 0.765(+/-0.004)for{'C':1,'gamma':10,'tol':0.0001} 0.765(+/-0.004)for{'C':1,'gamma':10,'tol':0.001} 0.765(+/-0.004)for{'C':1,'gamma':10,'tol':0.01} 0.765(+/-0.004)for{'C':10,'gamma':10,'tol':0.0001} 0.765(+/-0.004)for{'C':10,'gamma':10,'tol':0.001} 0.765(+/-0.004)for{'C':100,'gamma':10,'tol':0.0001} 0.765(+/-0.004)for{'C':100,'gamma':10,'tol':0.001} 0.765(+/-0.004)for{'C':1000,'gamma':10,'tol':0.0001} 0.765(+/-0.004)for{'C':1000,'gamma':10,'tol':0.001} 0.733(+/-0.067)for{'C':10,'gamma':1,'tol':0.0001} 0.733(+/-0.067)for{'C':10,'gamma':1,'tol':0.001} 0.733(+/-0.067)for{'C':10,'gamma':1,'tol':0.01} 0.733(+/-0.066)for{'C':1000,'gamma':1,'tol':0.0001} 0.733(+/-0.066)for{'C':1000,'gamma':1,'tol':0.001} 0.733(+/-0.066)for{'C':1000,'gamma':1,'tol':0.01} 0.732(+/-0.066)for{'C':100,'gamma':1,'tol':0.0001} 0.732(+/-0.066)for{'C':100,'gamma':1,'tol':0.001}	正解率: 0.765 適合率: 0.560 再現率: 0.018 混同マトリクス: [[2494 11] [759 14]]

処理内容	処理No	モデル/結果	テストデータによる結果
4を scoring='recall'として実行	5	GridSearchCV(cv=5, error_score=nan, estimator=SVC(C=1.0, break_ties=False, cache_size=200, class_weight='balanced', coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=1010, shrinking=True, tol=0.001, verbose=False), iid='deprecated', n_jobs=None, param_grid= {'C': [0.001, 0.01, 0.1, 1, 10, 100,1000], 'gamma' : [0.0001,0.001, 0.01, 0.1, 1, 10], 'tol':[0.0001, 0.001, 0.01]}, pre_dispatch='2*n_jobs', refit=True, return_train_score=False, scoring='recall', verbose=0) **********結果********** 正解率: 0.424, 適合率: 0.285, 再現率: 0.973 混同マトリクス:[[1307 3782] [42 1510]] [Best score]: 0.9613048438958615 [Best parameters]: {'C': 0.1, 'gamma': 1, 'tol': 0.0001} [Best estimator]: SVC(C=0.1, break_ties=False, cache_size=200, class_weight='balanced', coef0=0.0, decision_function_shape='ovr', degree=3, gamma=1, kernel='rbf', max_iter=-1, probability=False, random_state=1010, shrinking=True, tol=0.0001, verbose=False) [CV results]（上位 20 件） 0.961(+/-0.133)for{'C':0.1,'gamma':1,'tol':0.0001} 0.961(+/-0.133)for{'C':0.1,'gamma':1,'tol':0.001} 0.961(+/-0.133)for{'C':0.1,'gamma':1,'tol':0.01} 0.778(+/-0.185)for{'C':0.01,'gamma':0.1,'tol':0.0001} 0.778(+/-0.185)for{'C':0.01,'gamma':0.1,'tol':0.001} 0.777(+/-0.188)for{'C':0.01,'gamma':0.1,'tol':0.01} 0.564(+/-0.111)for{'C':0.1,'gamma':0.1,'tol':0.0001} 0.564(+/-0.111)for{'C':0.1,'gamma':0.1,'tol':0.001} 0.564(+/-0.111)for{'C':0.1,'gamma':0.1,'tol':0.01} 0.527(+/-0.264)for{'C':0.1,'gamma':0.01,'tol':0.0001} 0.527(+/-0.264)for{'C':0.1,'gamma':0.01,'tol':0.001} 0.527(+/-0.264)for{'C':0.1,'gamma':0.01,'tol':0.01} 0.525(+/-0.263)for{'C':10,'gamma':0.001,'tol':0.01} 0.523(+/-0.260)for{'C':10,'gamma':0.001,'tol':0.0001} 0.523(+/-0.260)for{'C':10,'gamma':0.001,'tol':0.001} 0.521(+/-0.288)for{'C':100,'gamma':0.0001,'tol':0.0001} 0.521(+/-0.288)for{'C':100,'gamma':0.0001,'tol':0.001} 0.521(+/-0.271)for{'C':1000,'gamma':0.0001,'tol':0.0001} 0.521(+/-0.271)for{'C':1000,'gamma':0.0001,'tol':0.001} 0.520(+/-0.290)for{'C':100,'gamma':0.0001,'tol':0.01}	正解率: 0.410 適合率: 0.281 再現率: 0.968 混同マトリクス: [[595 1910] [25 748]]

第 4 項 チューニングの結果

gbm では手順 5 により，正解率：0.800，適合率：0.590，再現率：0.490 が得られ，SVM では手順 5 により，正解率：0.410，適合率：0.281，再現率：0.968 が得られた．SVM は高い再現率であるが，適合率は低く，偽陽性が 7 割生じる．SVM による学習器を用いる場合，過度に低いインスリン投与量を推奨する可能性が高く，高血糖となりやすいと思われる．対して，gbm は半分程度の低血糖を予測できるが，擬陽性も 4 割生じる．何れを用いるか，あるいは使用するデータの変更（例えば，食事量，運動量がある），別の方法を検討する，更には開発を中止するかは，ベネフィット/リスクのバランスから判断することとなる．しかし，本書ではあくまで，事例紹介であり，このままサービスを提供するのではないので，gbm による学習器を用いた実装を行う．

第 5 節 実装

低血糖イベントの予測には，直近 10 回の空腹時血糖値とインスリン投与量のデータが必要である．そのため，スマートフォンなどのアプリとする場合，糖尿病手帳アプリに組み込み，手帳に記録されているデータを利用して，Regular insulin の投与量を変化させた場合の低血糖イベントの発生確率を表示する．

推定方法は，患者自身の直近の記録から log_Pre_blood_glucose_measurement, glucose_slp, glucose_sd, term, pRegular_ave, pglucose_ave, pglucose_sd, pNPH_ave, pUltraLente_ave を算出し，開発に用いた主成分分析の関数を用いて PC1，PC2，PC3，PC4，PC5，PC6 を算出する．Regular_insulin_dose は，0 単位から 10 単位又は過去の最大投与量+2 単位までの 1 単位に増加させた値を用いる．

ここでは，糖尿病手帳アプリに組み込むのではなく，機械学習のテストに用いたデータを用いて，実装した場合の例として 2 名分の出力イメージを図 4 -23 に示す．インスリン投与又は空腹時血糖測定ごとの過去の低血糖発生予測のグラフ上に，低血糖イベントがあった場合の Regular insulin 使用単位数を赤色マーカーで示し，黒い太線で今回の予測を示している．図 4 -23（1）は 8 単位以上の投与で低血糖イベントリスクが高くなり，似たパターンの過去 2 回は 16 単位の使用で低血糖イベントが生じていることが確認できる．図 4 -23（2）グラフは，総じて低血糖を生じやすい時と生じにくい時があり，今回はその中間のパターンとなっている．低血糖の多くは 15 単位以上で生じているが，今回は 7 単位以上で低血糖イベントリスクが高くなることが確認できる．いずれも，今回は低血糖イベントが生じやすそうであること，インスリンの量を減らすことでイベントを防止できる可能性があることを参考にインスリン使用量を調整できる．

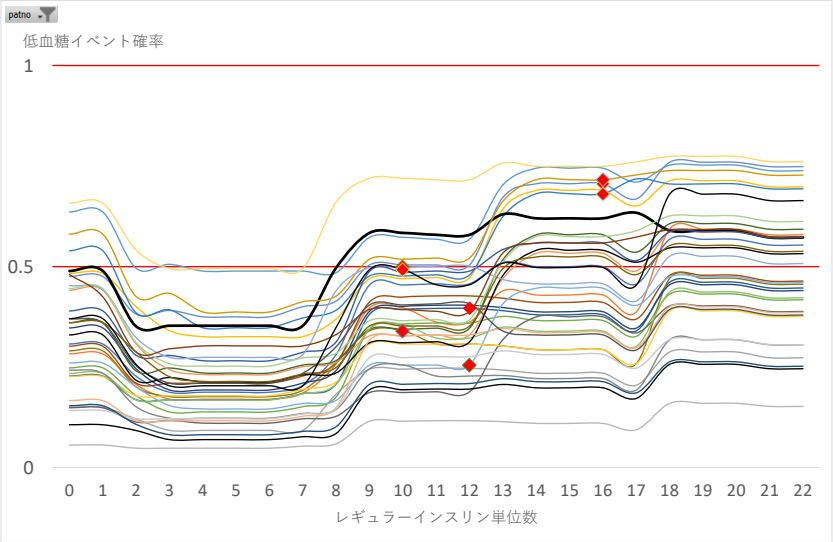


図 4 -23 予測結果表示のイメージ（1）

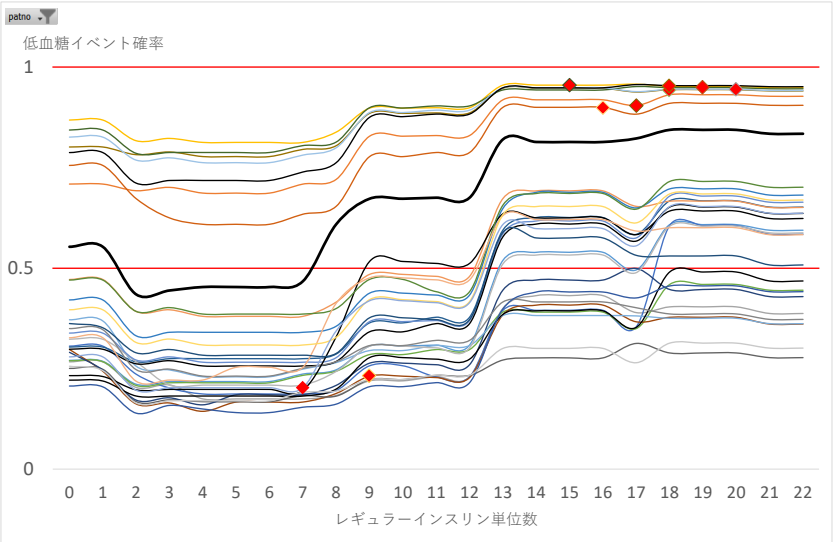


図 4 -23 予測結果表示のイメージ（2）

第 6 節 考察と留意事項

第 3 節第 2 項 B) の観測されたデータをほぼそのまま用いた場合 (model1) の学習では，svmRadial の再現率は 0，gbm は 0.36，　rf は 0.99（過学習と思われる）であったが，第 3 節第 2 項 C）の合成による特徴量の抽出（model2）や第 3 節第 2 項 D）のクラス変数の患者（patno）を連続変数による代替えと 1 回前の空腹時血糖値と regular_insulin 使用量の追加を行った (model3)，第 3 節第 2 項 E) の特徴量の絞り込み (model4) により，svmRadial の再現率は 0.15，gbm は 0.64 まで向上した．しかし，依然として「低血糖あり」のデータ

に対して学習器が正しく「低血糖あり」と判定する割合が低かったため，第 4 節第 4 節第 3 項では，model4 の特徴量を用いて，gbm と比較のため SVM の 2 手法に対してチューニングを実施した．gbm は過学習を防止するため，決定木の分岐に必要な最小データ数を制限して 3 つのパラメータを調整した結果，学習データでは，再現率は 0.55 であるが，適合率（陽性的中率）は 0.95 が得られた．SVM は標準化及び正則化の方法を変えた 3 パターンから標準化を用いて 3 つのパラメータについてチューニングを実施した結果，再現率は 0.97 と高いが，適合率は 0.28 となった．テストデータを用いた確認では，gbm の正解率は 0.80，再現率は 0.49，適合率は 0.59，SVM は 0.41， 0.99，0.28 となった．SVM は適合率，再現率のバランスが悪いが，gbm の適合率，再現率は不十分であるものの，ある程度バランスの取れたモデルを作成できたと考える．

しかし，SVM，gbm ともに想定していたよりも性能が向上しなかった．その理由として，血糖値の基本的な規定要因はインスリン投与量，投与時の血糖値，投与後の摂食量（主に炭水化物），運動量であるが，今回のデータでは食事量，運動量が定性的データであることに加え，欠損が多いため，インスリン投与量と血糖値のみを用いたことが考えられる．これ以外にも，Regular insulin を全く使用せず，NPH insulin でコントロールされている患者が含まれていたが，データ量との関係で除かずに使用しているなど，多少矛盾したデータも用いていることも原因の 1 つと考えられる．

より性能の良い学習器を得る方法として，学習データを増やす，学習に適したデータに絞る，特徴量を増やす方法が考えられる．学習データを増やす方法の 1 つである水増しとして SMOTE も試みたが（結果は未提示），重みづけの結果を超える性能は得られなかった．一般に，性能が向上するためには数倍から数百倍の学習データが必要になることが少なくないため，追加の学習データによる改善は容易ではないであろう．2 つ目の学習データに適したデータに絞るのは，今回の場合，インスリンを投与していないことが多い患者を除くなどが考えられるが，汎化性能が低くなる可能性が高い．3 つ目の特徴量を増やす方法は，元のデータには 19 変数あったが，使用したのは 7 変数である．10 変数にデータがある患者は 31 名，12 変数では 24 名と大きく減少するため有用ではないであろう．しかし，新規で学習データを取得するなら，特徴量に有用な測定項目を同時に増やすことで改善が可能かもしれない．その場合，例えば，食事の摂食量の定量的データがあればカーボ/インスリン比（即効型インスリン製剤 1 単位で処理できる糖質量）やインスリン効果値（即効型インスリン製剤 1 単位で補正できる血糖量）を算出して食事量をコントロールするカーボカウント法の考え方をモデルに取り込むことが可能である．このカーボカウント法は糖質量とインスリン量から摂食 4 時間後の理論上の血糖値が算出可能であるので，性能の大幅な向上が期待できる．従来，日本人の食事は炭水化物を含む料理が多く，炭水化物量の計算が困難という課題があったが，料理の写真を AI を用いて解析して炭水化物量，総カロリーなどを計算するアプリなどを利用することで，だれでも記録が可能となっている．

第 7 節 深層学習による血糖値の予測

「第 5 章第 1 節 学習済みモデルの利用」のセクション部分でも紹介しているように，再帰型ニューラルネットワーク（Recurrent Neural Network, RNN）は自然言語処理の分野で幅広く用いられている[30]．一方で，「時間」という概念を含む時系列データの予測にも活用されている．並び順に規則性，周期性及び特定のパターンがある時系列データを学習することで未知の時系列データが与えられたとき，そのデータの未来の状態予測を可能とする．

近年では，糖尿病患者が自身の血糖値を常にモニタリングできる持続血糖測定（Continuous Glucose Monitoring）機器が開発され，5～15 分間隔での血糖値を時系列データとして取得することも可能となっている[31,32,33]．こういった背景から，持続血糖測定データに対して RNN を用いた予測手法がいくつか報告されている[34,35,36]．これら手法の代表的なモデルが，Long Short-Term Memory（LSTM）及び Bidirectional - LSTM（Bi-LSTM）というニューラルネットワークである．LSTM は，通常の RNN では時系列データの短期的な予測はできるものの長期の依存関係を学習することが不可能（勾配消失問題）であった欠点を克服した手法である[37,38]．LSTM は名称の通り長期及び短期の依存性も学習できるモデルである．図 4 -24 のような基本構造を有しており，LSTM ブロックと呼ばれる回路を並べた仕組みになっている．通常の RNN に入力，忘却及び出力といったゲートを設けて長期的な依存性を学習するために情報を保存しておくことで，先の時間へ進み，必要になった時に，これら情報が活用されるのが LSTM の基本的な特徴である．

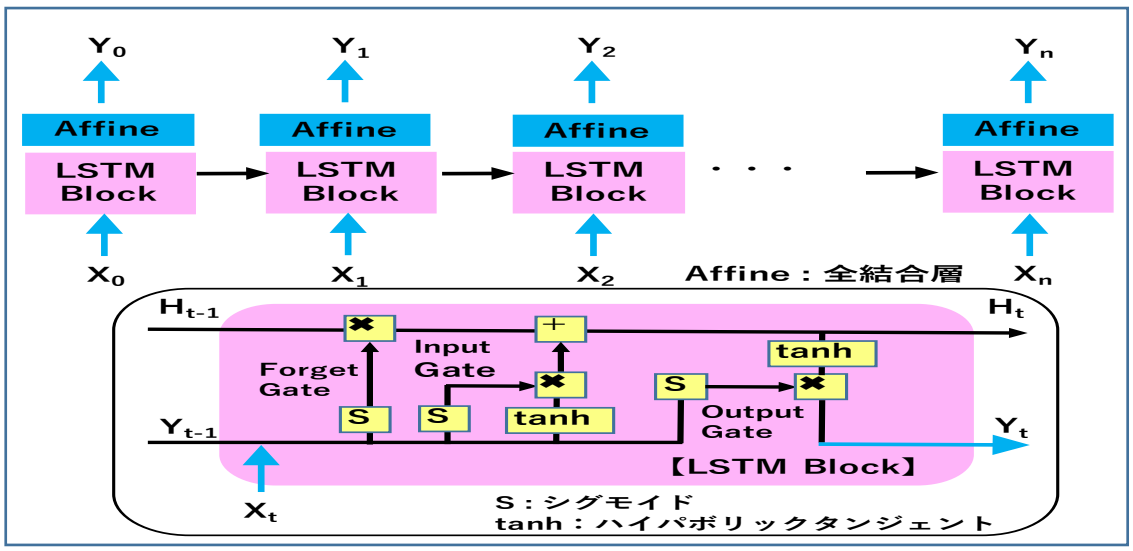


図 4 -24 LSTM の基本構造

LSTM は時刻 t-1 から時刻 t といったように過去から未来への 1 方向へ情報を伝播するモデルであるが，逆に未来から過去へ情報を伝播することも可能である．この「過去から」及び「未来から」の 2 方向の時系列の依存関係を考慮した LSTM が Bi-LSTM である[39]．Bi-LSTM は，図 4 -25 のような基本構造を有しており，双方向に情報が伝播するのが特徴

である．双方向で処理することにより一方向では見落とされる恐れのあるパターンを捕捉することが可能となる．

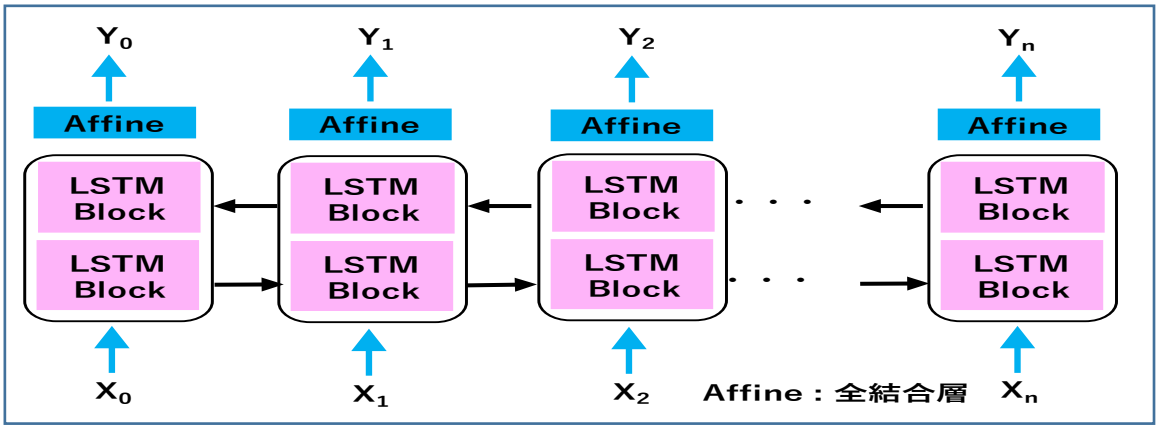


図 4 -25 Bi-LSTM の基本構造

血糖値の予測を可能とするオープンソースから入手可能な学習済みモデルが無かったため，転移学習やファインチューニングといった手法の適用が困難であった．そこで，「Diabetes Data Set」は，時系列データではあるが，持続血糖測定では無いことから予備的検討の位置づけで，上記の手法を参考にして深層学習（DL）によるモデル構築を行い，血糖値の予測を試みた．

第 1 項 本実装で用いた血糖値データ

「Diabetes Data Set」から血糖値のオブザベーションのみを抽出し，レコード数の多かった糖尿病患者 5 名（Patient ID：29,30,55,65,68）を血糖値予測のための解析用データとした．各糖尿病患者における血糖値のデータを日付順にソートして，レコード数を揃え，その総数は各糖尿病患者一人あたり 527 レコードであった．未来の血糖値の予測を解析の目的とした．このため，以下の図 4 -26 のようにランダムではなく時系列に学習データ及びテストデータに分割した．

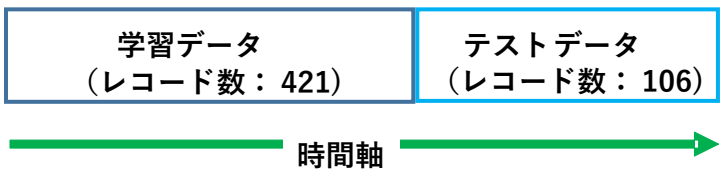


図 4 -26 データの分割方法

学習データの構造（例：Patient ID : 65,68）を図 4 -27 に示す．Gluc のカラムは血糖値，code のカラムは採血タイミングのコード値及び label は採血タイミングを表している．見出しの数字が患者 ID に相当する．行方向は時系列の繰り返しであり，日時順にソートしてい

るため，例えば行 3 から下 label が患者間で異なっている．テストデータの構造は学習データの構造と同一である．

	Gluc_65	code_65	label_65	Gluc_68	code_68	label_68
0	345	58	Pre-breakfast blood glucose measurement	134	58	Pre-breakfast blood glucose measurement
1	255	60	Pre-lunch blood glucose measurement	158	60	Pre-lunch blood glucose measurement
2	253	62	Pre-supper blood glucose measurement	258	62	Pre-supper blood glucose measurement
3	370	48	Unspecified blood glucose measurement	115	58	Pre-breakfast blood glucose measurement
4	95	58	Pre-breakfast blood glucose measurement	162	60	Pre-lunch blood glucose measurement
5	131	60	Pre-lunch blood glucose measurement	135	62	Pre-supper blood glucose measurement
6	178	62	Pre-supper blood glucose measurement	100	58	Pre-breakfast blood glucose measurement
7	164	48	Unspecified blood glucose measurement	148	60	Pre-lunch blood glucose measurement
8	201	58	Pre-breakfast blood glucose measurement	147	62	Pre-supper blood glucose measurement

図 4 -27 学習データの構造（例：Patient ID : 65,68）

各糖尿病患者の学習データにおける血糖値の推移及びタイミング別の採血回数を図 4 -28 及び図 4 -29 に示す．患者 ID:68 を除く糖尿病患者では，朝食前，昼食前及び夕食前の空腹時血糖及び随時血糖の測定回数はおおよそ 100 回であった．患者 ID:68 のみ随時血糖の測定は 17 回であった．また，いずれの糖尿病患者においても，食後血糖値のレコードは含まれていなかった．

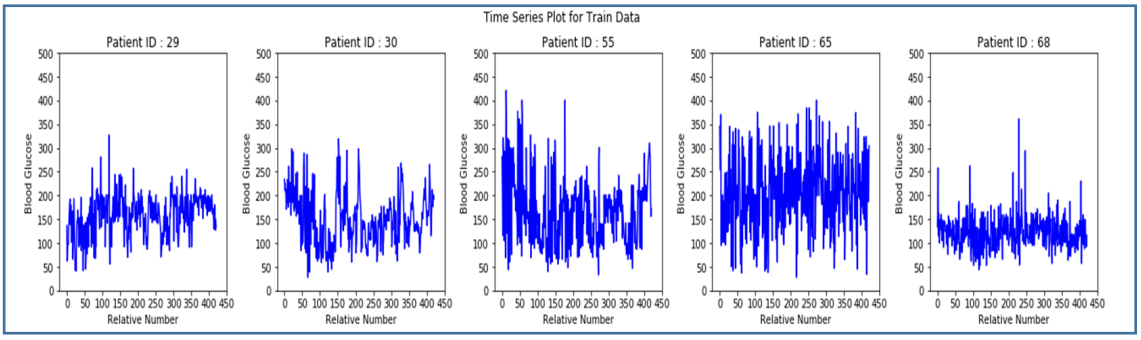


図 4 -28 学習データにおける血糖値の推移

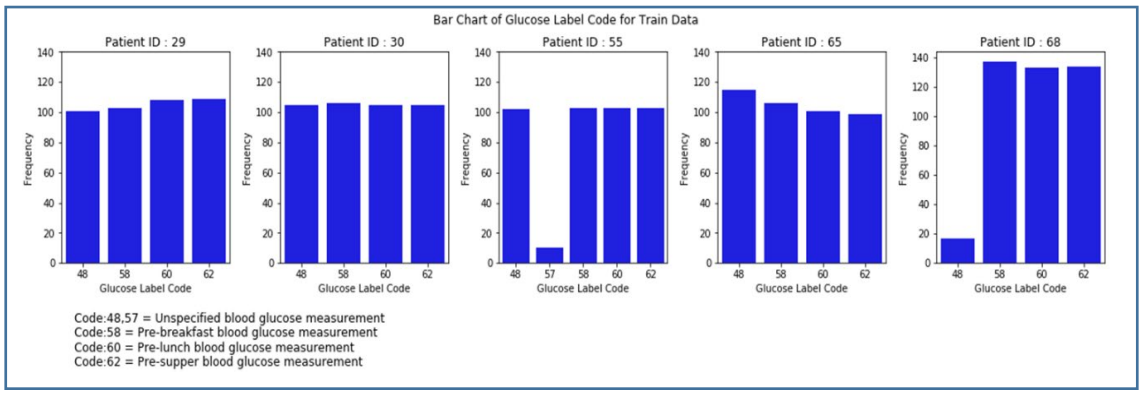


図 4 -29 学習データにおける採血タイミングごとの回数

各糖尿病患者のテストデータにおける血糖値の推移及び血糖採取タイミングごとの回数を図 4 -30 及び図 4 -31 に示す．朝食前，昼食前及び夕食前の空腹時血糖及び随時血糖の測定回数はおおよそ 30 レコード前後であった．学習データと同様に，いずれの糖尿病患者においても，食後血糖値のレコードは認められなかった．患者 ID:68 については，学習データとテストデータとの間で，相対的に随時血糖の測定回数に乖離があった．

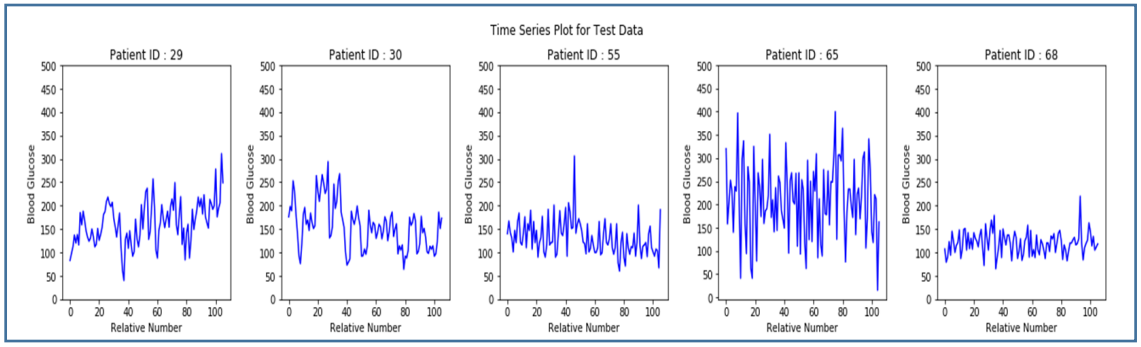


図 4 -30 テストデータにおける血糖値の推移

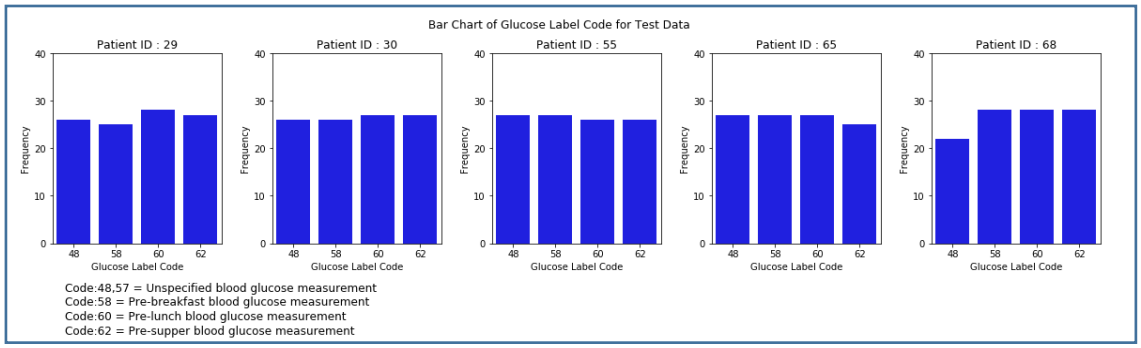


図 4 -31 テストデータにおける血糖採取タイミングごとの回数

なお，血糖値は，各糖尿病患者に正規化変換を施し，深層学習によるモデル構築の検討には，この変換したデータを用いた．そして，予測値を算出する際に，元スケールに戻した．本実装では，以下の式に従い，最大値を 1，最小値を 0 とした正規化を行った．

$$Xpost = \frac{Xpre - Xmin}{Xmax - Xmin}$$

$Xpost$: 変換後の値，$Xpre$: 実測値
$Xmax$: 最大値，$Xmin$: 最小値

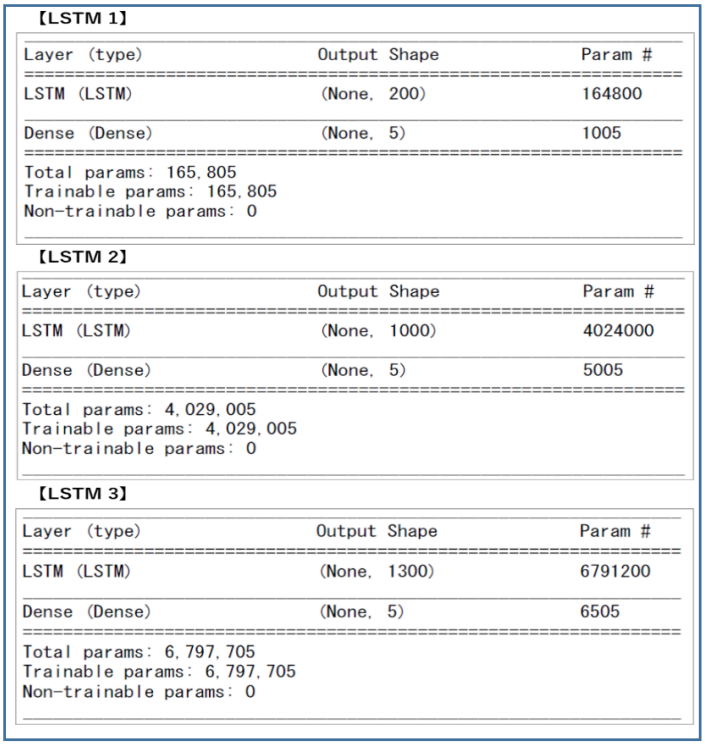
第 2 項 モデル構築

血糖値の予測を行うためのモデルは，表 4 -28 に示した RNN モデルを検討した．出力層の活性化関数は，いずれのモデルも Linear とした．LSTM 及び Bi-LSTM のモデルのサマリーを図 4 -32 及び図 4 -33 に示す．

表 4 -28 検討した RNN モデル

	Layer	Model Name
LSTM	200	LSTM 1
	1000	LSTM 2
	1300	LSTM 3

	Layer	Model Name
Bi-LSTM	200	Bi-LSTM 1
	800	Bi-LSTM 2
	1000	Bi-LATM 3



【LSTM 1】

Layer (type)	Output Shape	Param #
LSTM (LSTM)	(None, 200)	164800
Dense (Dense)	(None, 5)	1005

Total params: 165,805
Trainable params: 165,805
Non-trainable params: 0

【LSTM 2】

Layer (type)	Output Shape	Param #
LSTM (LSTM)	(None, 1000)	4024000
Dense (Dense)	(None, 5)	5005

Total params: 4,029,005
Trainable params: 4,029,005
Non-trainable params: 0

【LSTM 3】

Layer (type)	Output Shape	Param #
LSTM (LSTM)	(None, 1300)	6791200
Dense (Dense)	(None, 5)	6505

Total params: 6,797,705
Trainable params: 6,797,705
Non-trainable params: 0

図 4 -32 LSTM のモデルサマリー

【Bi-LSTM 1】

Layer (type)	Output Shape	Param #
Bi-LSTM (Bidirectional)	(None, 400)	329600
Dense (Dense)	(None, 5)	2005

Total params: 331,605
Trainable params: 331,605
Non-trainable params: 0

【Bi-LSTM 2】

Layer (type)	Output Shape	Param #
Bi-LSTM (Bidirectional)	(None, 1600)	5158400
Dense (Dense)	(None, 5)	8005

Total params: 5,166,405
Trainable params: 5,166,405
Non-trainable params: 0

【Bi-LSTM 3】

Layer (type)	Output Shape	Param #
Bi-LSTM (Bidirectional)	(None, 2000)	8048000
Dense (Dense)	(None, 5)	10005

Total params: 8,058,005
Trainable params: 8,058,005
Non-trainable params: 0

図 4 -33 Bi-LSTM のモデルサマリー

また，血糖値の予測は，予測するポイントの前 10 ポイントを学習させて次ポイントの血糖値を予測させた．患者 ID : 29 を例とした血糖値予測のための学習イメージを図 4 -34 に示す．

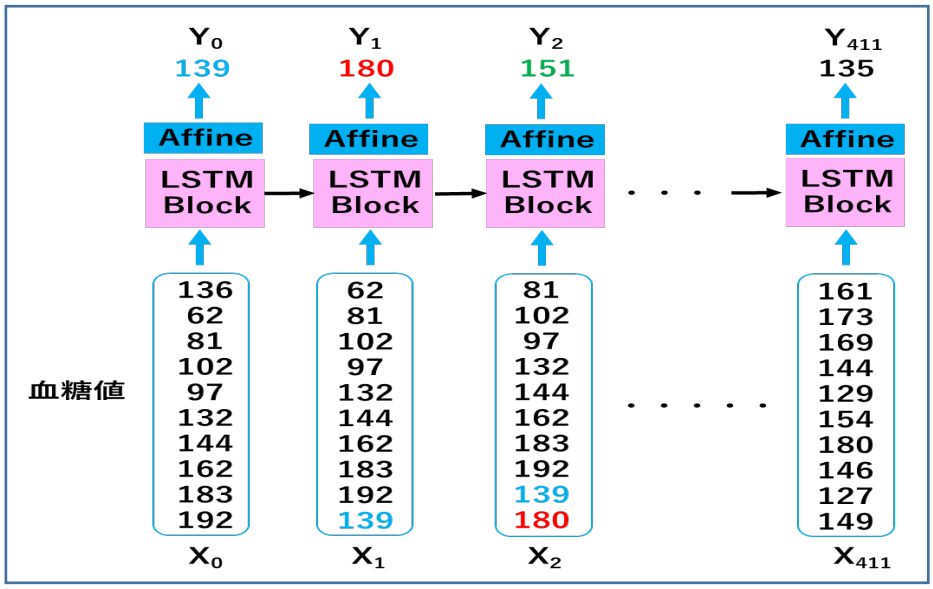


図 4 -34 血糖値予測のための学習イメージ（例：Patient ID : 29）

第 3 項 実装結果

学習データのうち 20%を検証データとしてモデル評価及びパラメータチューニングに利用した．LSTM 及び Bi-LSTM の学習過程を図 4 -35 及び図 4 -36 に示す．なお，学習結果は早期中止（early stopping）を考慮した損失値（mean squared error）で評価した．LSTM 及び Bi-LSTM のいずれのモデルもエポックが進むに連れて収束が認められた．最終エポックにおける学習データ及び検証データの損失値に大きな乖離は認められなかった．

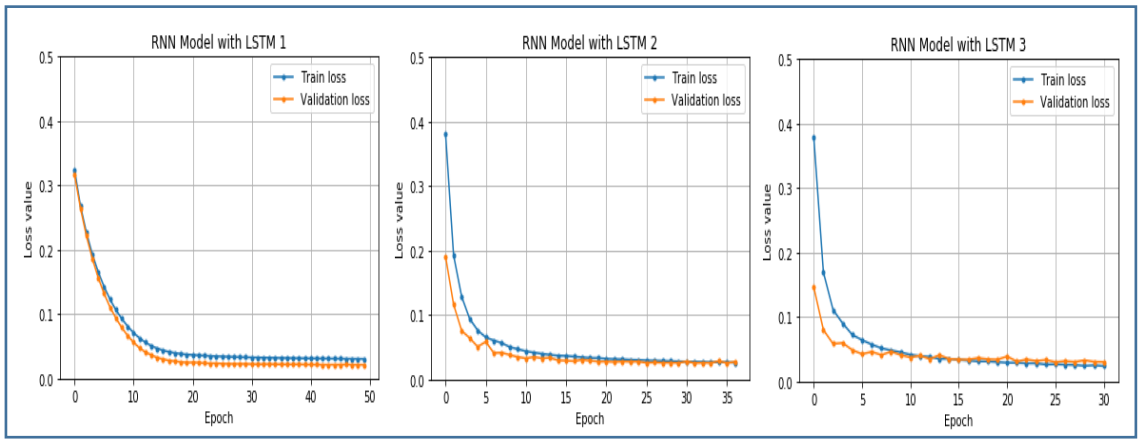


図 4 -35 LSTM の学習過程

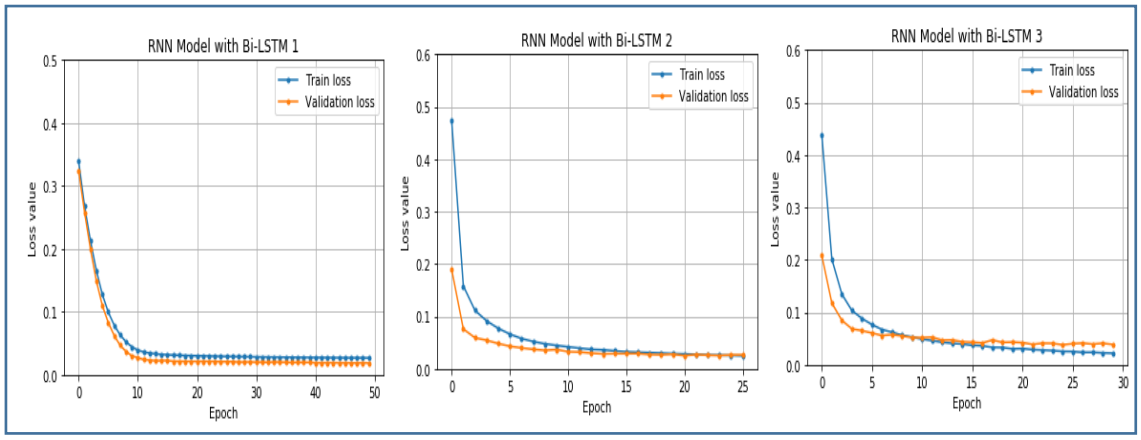


図 4 -36 Bi-LSTM の学習過程

患者 ID：29 を例とした学習データにおける血糖実測値及び各 RNN モデルから推定した血糖予測値の時系列プロットを図 4 -37 及び図 4 -38 に示す．LSTM 及び Bi-LSTM のいずれのモデルも中間層が浅いほど血糖推移パターンをうまく捉えられていなかった．

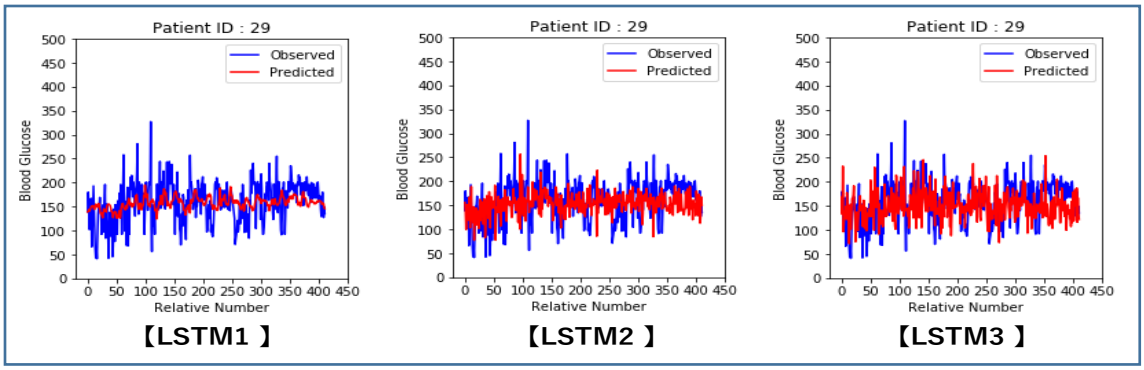


図 4 -37 学習データにおける実測値と LSTM に基づく予測値の時系列プロット

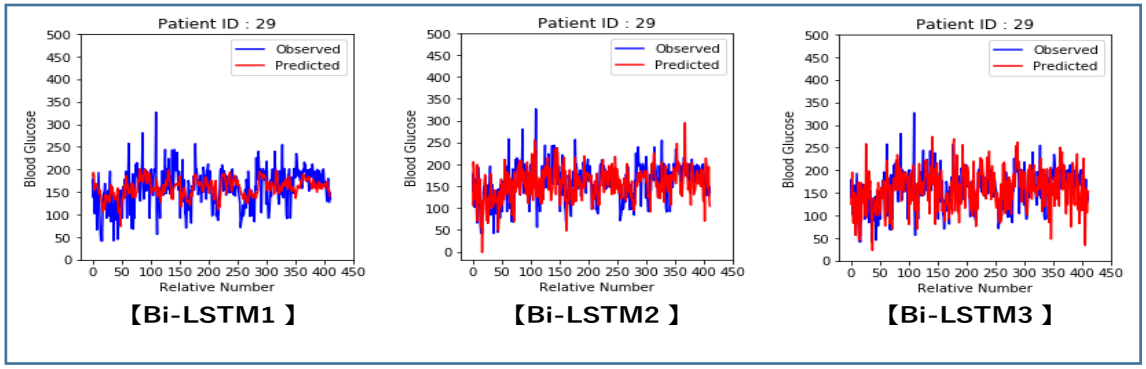


図 4 -38 学習データにおける実測値と Bi-LSTM に基づく予測値の時系列プロット

患者 ID：29 を例としたテストデータにおける血糖実測値及び各 RNN モデルから推定した血糖予測値の時系列プロットを図 4 -39 及び図 4 -40 に示す．LSTM 及び Bi-LSTM のいずれのモデルも学習データの結果と同様な傾向であった．

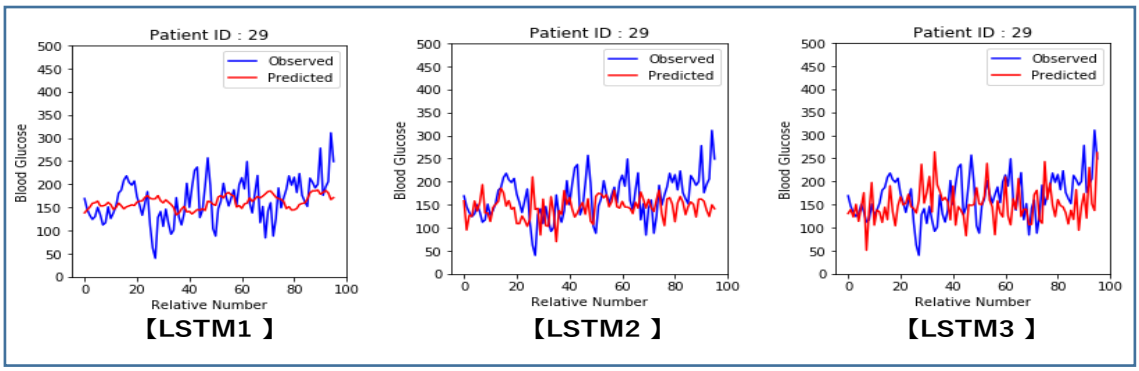


図 4 -39 テストデータにおける実測値と LSTM に基づく予測値の時系列プロット

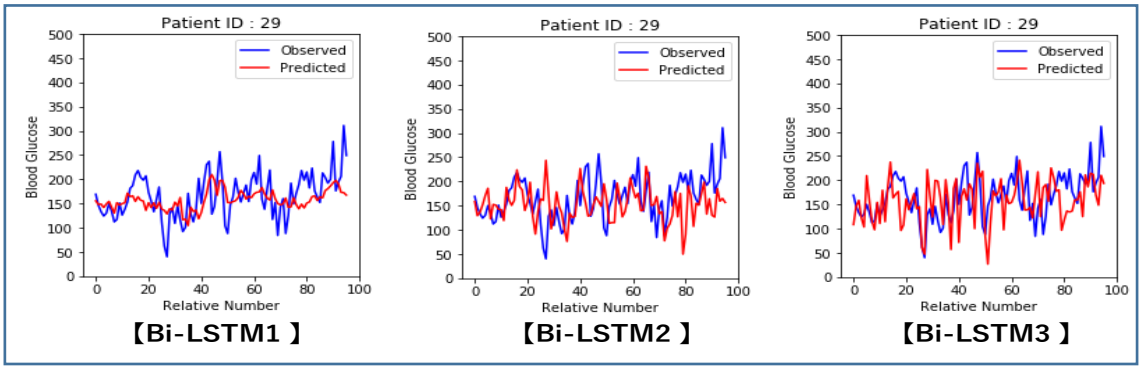


図 4 -40 テストデータにおける実測値と Bi-LSTM に基づく予測値の時系列プロット

各 RNN モデルの性能評価指標は RMSE（Root Mean Squared Error）とした.

$$RMSE = \sqrt{\frac{\sum_i \left(y_{obs,i} - y_{pred,i}\right)^2}{n}}$$ $y_{obs,i}$: 観測値, $y_{pred,i}$: 予測値 ともに元スケール

学習データ及びテストデータにおける性能評価結果を表 4 -29 に示す. 学習データにおいては, LSTM 及び Bi-LSTM のいずれのモデルも中間層の深さに関わらず予測誤差に大きな差異はなかった. しかしながら, 実測値と予測値の時系列プロットから, 中間層が浅いほど血糖推移パターンをうまく捉えられていなかった. また, 学習データに比べてテストデータでは中間層が深いほど予測誤差が大きく過学習が示唆された.

実測値と予測値の時系列プロットといった視覚的評価及び RMSE の結果を考慮すると LSTM 2 又は Bi-LSTM 2 がリーズナブルな血糖予測モデルと考えられた. 論文[32,33,34]での RMSE は概ね 10～30 程度であるが, LSTM 2 より得られた RMSE は, 37.18～88.15 及び Bi-LSTM 2 より得られた RMSE は 37.84～99.58 であったこと並びに実測値と予測値の時系列プロットといった視覚的評価も含め, 既存方法に迫る予測精度は得ることはできなかった. この理由としては, 更なるモデルの検討の必要性もあるが, 本実装で使用した「Diabetes Data Set」は, 周期性を持った時系列データではあるものの, 持続血糖測定ではなく, 食後血糖値のレコードも存在しない食前の空腹時血糖を中心としたスポット的なデータであったというデータの質的な側面が考えられる.

表 4 -29 RNN の性能評価（RMSE）

	RNN Model	Patient ID : 29	Patient ID : 30	Patient ID : 55	Patient ID : 65	Patient ID : 68
学習用データ	LSTM 1	42.03	52.31	69.76	77.09	35.48
	LSTM 2	42.82	50.09	64.96	71.62	38.05
	LSTM 3	41.87	48.24	64.39	66.49	39.29
	Bi-LSTM 1	39.89	44.19	63.84	76.33	36.23
	Bi-LSTM 2	40.50	46.13	64.74	75.31	43.01
	Bi-LSTM 3	42.80	44.16	69.02	70.92	45.50
テスト用データ	LSTM 1	48.49	48.86	38.72	83.10	29.77
	LSTM 2	55.62	55.78	47.15	88.15	37.18
	LSTM 3	61.93	57.56	46.74	98.92	37.75
	Bi-LSTM 1	43.09	45.43	41.31	84.99	33.93
	Bi-LSTM 2	56.39	49.82	49.15	99.58	37.84
	Bi-LSTM 3	53.73	73.32	68.84	110.60	39.61

本章の終わりに

本章では，臨床試験データや RWD に基づく判別・識別，予測に機械学習の手法を用いる場合を想定して，空腹時血糖値などのデータから低血糖イベントの予測を機械学習によって行い，そうしてできたモデルをシミュレーションに用いることでインスリンの投与量と低血糖イベントリスクを可視化した．利用できるデータ（特徴量）が限られているため，予測精度は十分とは言えないが，構造化データから機械学習に必要な特徴量の作り方・取り出し方については，実際の試行錯誤を含めて丁寧に紹介した．この部分はデータに関する知識を機械学習に組み込む部分でもあるので，実際に機械学習を用いる分野，データによって大きく異なる分野ではあるが，時系列の繰り返しデータ，“患者”という通常は定量化されない情報を学習データに含まれない患者への拡張可能な数値化を行う方法は参考になると考える．

機械学習についても，手法の選択とチューニングも細かく行い，紹介している．機械学習を用いる目的，必要な性能によるが，検討する手法は 2〜5 種類とするのが効率がよいであろう．また，手法の選択のために第 3 節第 2 項では複数の手法をまとめて検討する事例を示すため，パラメータの自動調整を行っているが，時間がかかるので，チューニングを行わず，デフォルトのパラメータを用いてもよいであろう．

また，道具としてだれでも簡便に機械学習を利用することは，本報告書のテーマであるので，このような丁寧な手法の選択やチューニングを行うのではなく，自動化された機械学習ソフトウエアを用いた場合について紹介する．ソフトウエアの宣伝ではないので，商品名は控えるが，有名な機械学習を自動で行うソフトウエアに特徴量の加工を行った第 3 項と同じデータを投入した結果，xgbTree，LightBGM，ExtraTrees，ENET Blender など 8 種類の機械学習の手法のオプションの違いにより延べ１３種類が実行され，RandomForest Classifier (Entropy)が推奨手法となった．その結果の概要を表 4 -30 に示す．赤字はデフォルトから変更されているパラメータで，bootstrap と criterion 以外の 5 種類のパラメータがチューニングにより調整されているが，不均衡データへの対象である重みづけ（class_weight）は行われていない．特徴量の役割の図 4 -41 の上位 5 つのうち２つは図 4 -22 GBM 手順 5 のモデルでの特徴量の重要度の上位 5 つに含まれておらず，特徴量の扱いに違いが生じている．結果は第 3 項の gbm の結果に比べて正解率は若干悪くなっているが，多くのパラメータがチューニングされたことによるのか，再現率は 0.16 良い結果（代わりに特異度は 0.09 低下）となっており，簡便さ，性能ともに有用であると考えられた．

表 4 -30 自動機械学習ソフトの結果

<table>
<tr><th>Best parameters</th><th>結果（テストデータに対する）</th></tr>
<tr><td>balanced_bootstrap:False,
bootstrap:False, class_weight:None,
criterion:entropy, max_depth:None,
max_features:0.4, max_leaf_nodes:200,
min_samples_leaf:5,
min_samples_split:10, n_estimators:500,
random_state:1234, replace:True,
subsample:1.0</td><td>正解率：0.7655
適合率：0.4983
再現率：0.6466
混同マトリクス：
<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">予測</td></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td rowspan="2">実測</td><td>0</td><td>2033</td><td>503</td></tr>
<tr><td>1</td><td>273</td><td>499</td></tr>
</table></td></tr>
</table>

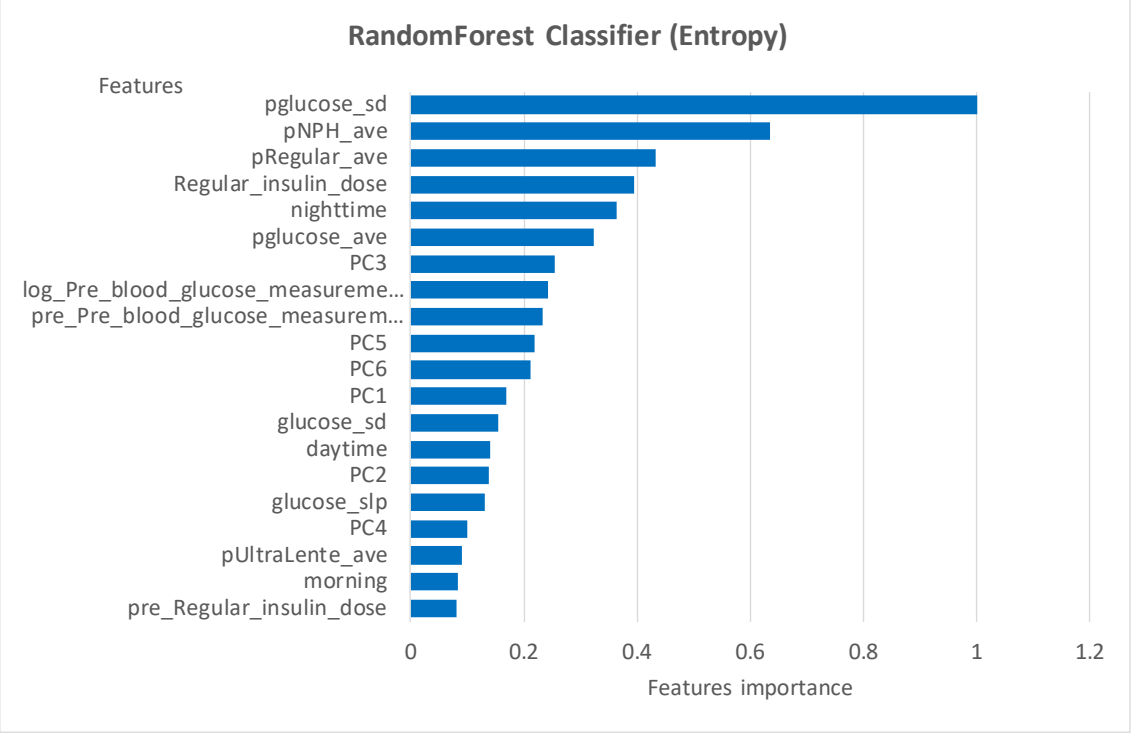


図 4 -41 自動学習ソフトによる特徴量の重要度

第５章　学習済みモデルの利用に関する解説・実用例の紹介

医療や医薬品開発の分野への深層学習（DL）の利用が注目されている[40,41,42,43]．深層学習は，ディープラーニングとも呼ばれる機械学習に含まれる技術で，ニューラルネットワークを拡張し，高精度の分析を可能にした手法である[44]．「AI ってなに？」[1]では，X 線画像から胸水の有無を判別する深層学習を行い，紹介したが，11 万枚の画像を用いても十分な性能は得られなかった．深層学習は第 4 章第 3 節で用いたような他の機械学習手法と異なり特徴量の抽出から学習するため，一般的に多量のデータが必要となる．そのため，製薬企業が自社の課題解決を目的として，新たな深層学習による AI を開発することは，必要な量と品質の開発データの確保が大きな障害となると思われる．そこで，本書では，新たな深層学習を行う場合は扱わず，既存の深層学習によるモデルを利用する方法を紹介する．

第 1 節 学習済みモデルの利用

「AI って何？」でも述べたように，入力層，中間層（又は隠れ層）及び出力層から構成され，中間層を追加して，計 4 層以上に階層を深くしたニューラルネットワークによる学習を深層学習と呼ぶ[注 45]．中間層に工夫を加えることで，画像認識，自然言語処理及び音声認識といった様々な分野において，人間がプログラミングにより教えなければならなかった論理的特徴や言語で説明困難な特徴をコンピュータが自ら学習し，性能を向上させることが可能となっている[44]．深層学習には，画像認識，自然言語処理及び音声認識など非構造データから学習によって得られた特徴を抽出するモデルを再利用できるといった利点もある．通常の機械学習は特徴量を人が加工や選別するのに対して，深層学習では，特徴量の抽出もニューラルネットワークが行うため，より膨大な開発データが必要となる．しかし，現実的には目的に見合った実用水準を満たす十分な量のデータ収集，教師付き開発データへの成型及びモデルの構築が必要となり，費用，人的資源及び時間といった面で困難な場合もある[45,46]．こうした課題への対処として，ゼロからモデルを構築するのではなく，関連したタスク[注 46]にて学習した「学習済みモデル（Pre-Trained model）の利用」がスタンダードとなっている．更に，ターゲットとなるデータで学習済みモデルの微調整を行い，新しいタスクの識別に活用する「転移学習[注 47]（Transfer Learning）」や「ファインチューニング[注 48]（Fine-Tuning）」といったアプローチもある．

注45 深層学習の基本的な説明，他の機械学習との関係，活性化関数などは「AI ってなに？」を参照．

注46 この章におけるタスクとは，問題や質問，および利用可能なデータに基づいて機械学習により行われる予測または推論の種類．

注47 ある問題を効果的かつ，効率的に解くために別の関連した問題（ソースドメイン）の学習結果を利用し，より高い予測精度を得る手法である．詳細は，第 3 節を参照．

注48 学習済みモデルの一部の重みを学習させ，目的とするタスクへ適合させる手法である．転移学習の手法の一つとして捉えることもできる．詳細は，第 4 節を参照．

画像認識の分野においては，主に ImageNet 注 49の一部を利用した ILSVRC（大規模画像コンペ）において 120 万の学習画像，5 万の検証画像及び 10 万のテスト画像にて 1000 クラスの識別を学習し，人の認識精度に迫る又は超える性能が報告されている畳み込みニューラルネットワーク（Convolutional Neural Network, CNN）が公開されており，学習済みモデル[47,48,49]として利用できる．更に，敵対的生成ネットワーク（Generative Adversarial Network, GAN）に基づく画像生成や CNN に基づくオブジェクト検出における学習済みモデル[50,51,52]もある．

自然言語処理の分野においては，単語埋め込み（Word Embedding）という浅いニューラルネットワークを利用して単語を実数ベクトルで表現するニューラル言語モデル[53,54,55,56]，構成性，多義性，照応性，周期的な依存関係，合意，否定といった複雑な自然言語処理に対応した，より深いニューラルネットワークである再帰型ニューラルネットワーク（Recurrent Neural Network, RNN）モデル[57,58]及び RNN とは異なり，再帰的構造を有さない Attention メカニズムに基づくニューラルネットワーク言語モデル[59,60]も登場し，事前に学習した言語モデルを使用することにより広範囲な自然言語処理評価で最先端の結果が見出されている．

音声認識の分野では，これまで音響モデルとして用いられていた隠れマルコフモデル（Hidden Markov Model, HMM）及び HMM からの特徴抽出として多次元混合ガウス分布（Gaussian Mixture Model, GMM）が用いられてきた．この GMM の代わりにディープニューラルネットワークが用いられ，大語彙連続音声認識タスク注 50において精度の改善が認められている[61,62,63]．CNN に基づく音声認識，RNN に基づく音声言語変換及び時系列データに対して CNN を導入した音声生成といった学習済みモデル[64,65,66]もある．

「AI って何？」で紹介したように，クラウドベンダーから AI プラットフォームがリリースされており，学習済みモデルを API で使用可能なサービスが提供されている．クラウドベンダーから提供されている API サービスを活用すれば，手軽に学習済みモデルを用いて機械学習や深層学習を実行することができる．ベンダーにより提供している機能や種類など特徴が異なるが，概ね，画像認識，自然言語処理及び音声認識が中心の汎用的なものである．

ある種の疾患の診断や医薬品開発への応用など，特化した目的の達成を目指すケースでは，学習済みモデルを用いて転移学習やファインチューニングといった手法を駆使した独自開発を考慮する必要性がある．TensorFlow Hub（https://www.tensorflow.org/hub/）や Model Asset eXchange（https://developer.ibm.com/exchanges/models/）といった転

注49 概念体系に基づいて収集，整理された大規模画像データセット．http://www.image-net.org/

注50 大語彙は多くの単語を用いているという意味であり，制約のない一般の音声を連続的に認識する技術で，音声翻訳や音声からの文書作成といったサービスに応用されている．「PDA をクライアント端末としたセンターサーバーによる大規模語彙の連続音声認識処理システム」https://www.medtronic.com/jp-ja/about/news/pressrelease/2018-12-03.html．「音声エージェント機能 しゃべってコンシェル」https://www.nttdocomo.co.jp/info/news_release/2012/02/27_00.html.

移学習の各種タスクで再利用が可能な学習済みモデルのオープンソースライブラリも公開されている[注51].

第 2 節 学習済みモデルの紹介

第 1 項 医療分野における学習済みモデルの利用

A） 画像認識に用いられる学習済みモデル

◆ 畳み込みニューラルネットワーク

医療分野における画像認識に用いられる学習済みモデルの説明の前に，コンピュータが画像を情報として処理できるように数値に変換する方法として，画像認識分野において広く応用されている畳み込みニューラルネットワーク（CNN）を紹介する．CNN は，人間が持つ視覚野の神経細胞の働きを模したネオコグニトロン[注52]が大元となっており，現在までに多くの改良が重ねられている[67].

CNN は，図 5 -1 に示すように，入力層，畳み込み層とプーリング層を繰り返した中間層，全結合層及び出力層を基本構造としている．

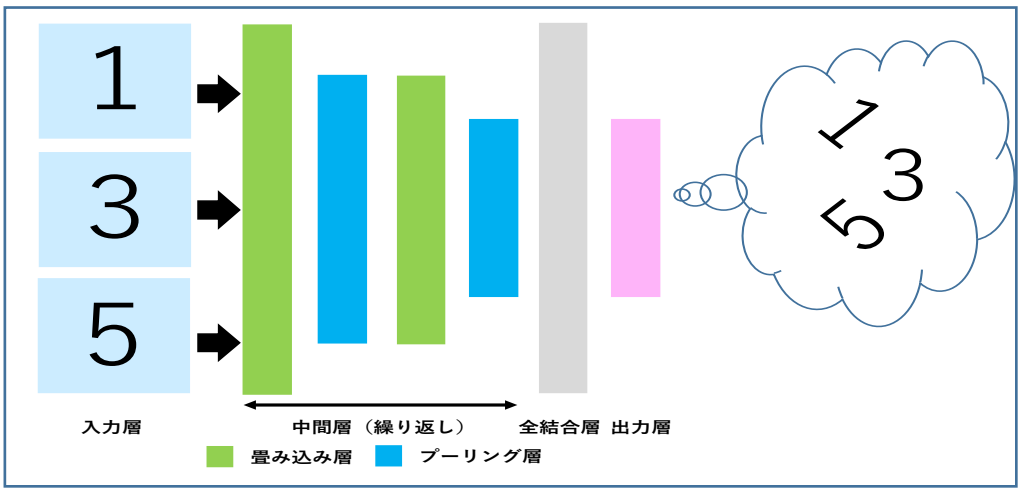


図 5 -1 CNN の基本構造

画像データは，数値データと異なりピクセルが並んだデータ構造を有しており，図 5 -2 ように，モノクロ画像なら各ピクセルに 1 つ，カラー画像なら RGB の 3 つの値（チャンネル）として色情報を持つ．なお，RGB とは，赤，緑，青の三つの原色を混ぜて幅広い色調を再現する表現法の一種である．R, G, B のそれぞれの濃度を 256 階調（0～255）の整数値で表

注51 章末の表 5 -12 【代表的なフレームワークと学習済みモデル】を参照.

注52 福島邦彦によって考えられた神経細胞を模倣した S 細胞層（画像の濃厚パターンを検出）と C 細胞層（物体の位置が変動しても同一物体と認識）を交互に複数組み合わせたモデル.

す．したがって，入力層における画像データは，数値情報として縦，横及びチャンネルの 3 次元配列で表示される．

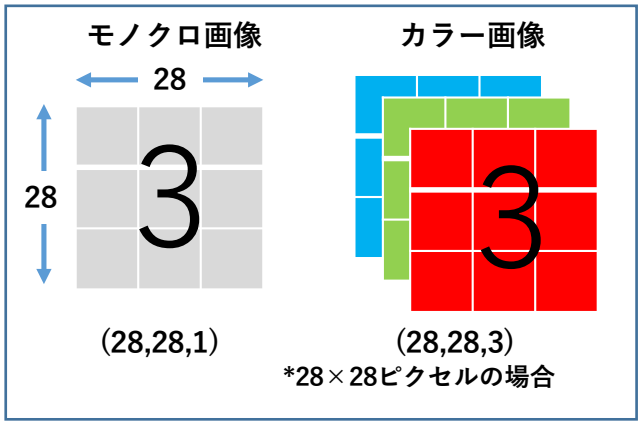


図 5 -2 画像データの数値情報

例として，28×28 ピクセルで，数字の 3 を画像入力した場合には，図 5 -3 に示すような数値情報となり，色の濃淡（黒：0～白：255）に応じて数値が大きくなっている．なお，画像データの数値情報を見やすくするため，正規化[注 53]せずにモノクロ画像データを用いて CNN を説明する[注 54]．

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	38	43	105	255	253	253	253	253	253	174	6	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	43	139	224	226	252	253	252	252	252	252	252	252	158	14	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	178	252	252	252	252	253	252	252	252	252	252	252	252	59	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	109	252	252	230	132	133	132	132	189	252	252	252	252	59	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	4	29	29	24	0	0	0	0	14	226	252	252	172	7	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	85	243	252	252	144	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88	189	252	252	252	14	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91	212	247	252	252	252	204	9	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	32	125	193	193	193	253	252	252	252	238	102	28	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	45	222	252	252	252	252	253	252	252	252	177	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	45	223	253	253	253	253	255	253	253	253	253	74	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	31	123	52	44	44	44	44	143	252	252	74	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	252	252	74	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86	252	252	74	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	5	75	9	0	0	0	0	0	0	0	98	242	252	252	74	0	0	0	0	0	0
20	0	0	0	0	0	0	61	183	252	29	0	0	0	0	18	92	239	252	252	243	65	0	0					
21	0	0	0	0	0	0	208	252	252	147	134	134	134	134	203	253	252	252	188	83	0	0	0					
22	0	0	0	0	0	0	208	252	252	252	252	252	252	252	252	253	230	153	8	0	0	0	0					
23	0	0	0	0	0	0	49	157	252	252	252	252	252	217	207	146	45	0	0	0	0	0	0					
24	0	0	0	0	0	0	0	7	103	235	252	172	103	24	0	0	0	0	0	0	0	0	0					
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					

図 5 -3 画像データにおける数値情報の具体例

注53 ピクセル値を 0～1 の範囲に収める処理．各ピクセル値を 255 で割ることによって 0～1 の範囲に収めることができる．

注54 手書き文字のデータは，Python ライブラリの keras に含まれている MNIST データを利用した．

入力層における情報は，畳み込み層に送られ，フィルタを用いて画像から特徴量を抽出する．このフィルタは，通常，3×3 範囲のピクセル（9 ピクセル）など画像よりも小さいサイズが用いられている．畳み込みとは，このフィルタを画像の左上から順次重ねていき，画像とフィルタをそれぞれ掛け合わせた総和を取った値を求めていく処理となる．具体的には，先述の 3 の画像情報のうち，ピンクで囲った部分を用いて，畳み込み処理を行うと図 5 -4 に示したように 477 という値に集約される[注 55]．

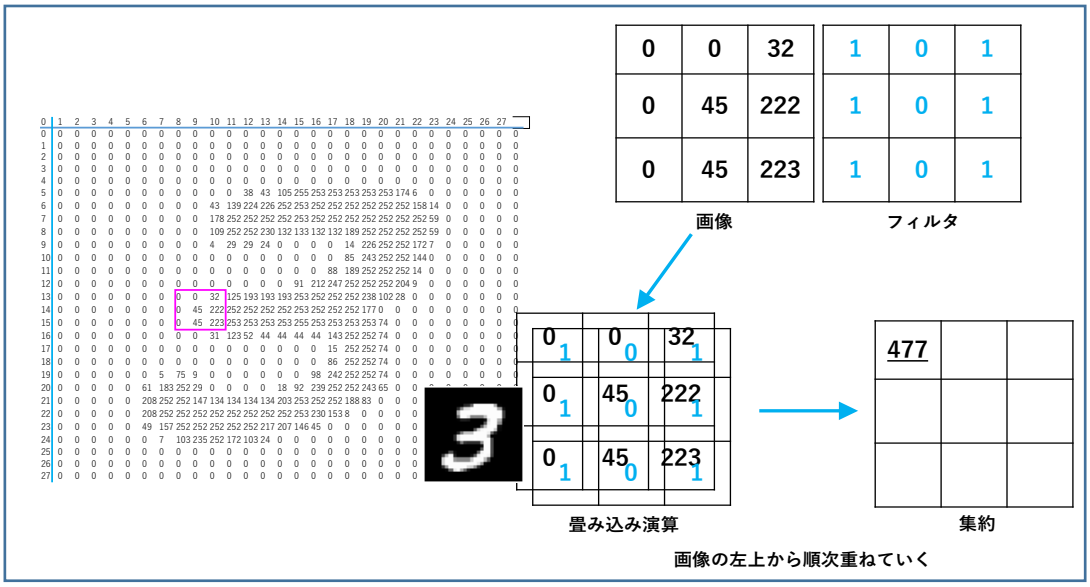


図 5 -4 畳み込み層における処理

このように畳み込みにより得られた新しい数値情報を特徴マップと呼ぶ．CNN では，このフィルタの値をどういった値にすれば誤差が小さくなるか学習していくことになる．よって，フィルタの各値が通常のニューラルネットワークの重みに相当する．この畳み込み処理は，人間の視覚野が持つ局所受容野に対応しており，移動不変性[注 56]の獲得に貢献する[67]．すなわち，畳み込みによって，「位置のズレ」に強いモデルができることになる．

畳み込み層では，モデルパラメータであるフィルタの学習を行ったが，プーリング層は，決められた演算を行うだけである．プーリング層では，図 5 -5 に示したように，特徴マップの最大値を求める処理が行われ，この処理は，マックスプーリングと呼ばれる．最大値を求める以外に平均値を求める平均プーリングもある．

注55 計算例として分かり易くするため，単純に画像とフィルタをそれぞれ掛け合わせた総和を取った値を示したが，実際には，バイアスを加えて ReLU（ランプ関数），Tanh（ハイパボリックタンジェント）といった活性化関数を施した値が次の層へ出力される．

注56 画像全体をフィルタがスライドすることで，特徴がどこにあっても抽出できること．

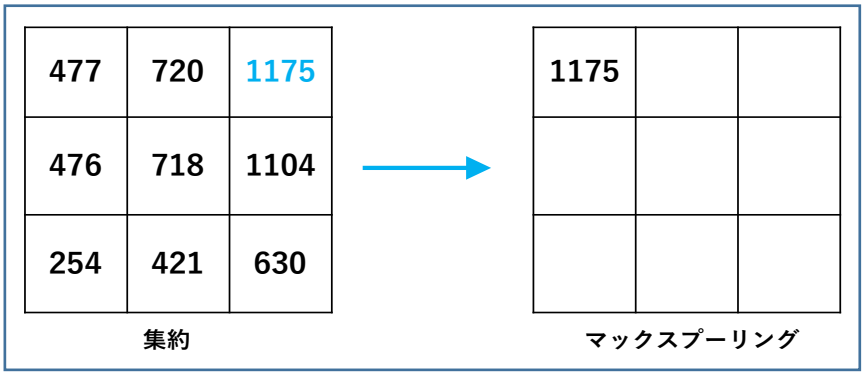


図 5 -5 プーリング層における処理

この畳み込み処理とプーリング処理を繰り返すことで，画像データにおける数値情報が集約され，順次，画像の特徴量が抽出されていく．この事例の場合では，最終的に数字画像の識別が目的であるため，通常のニューラルネットワークによる回帰や分類と同様に出力を 1 次元にする必要がある．そこで，図 5 -6 のように多次元情報を一次元に平坦化して，出力層へ接続するのが全結合層である．

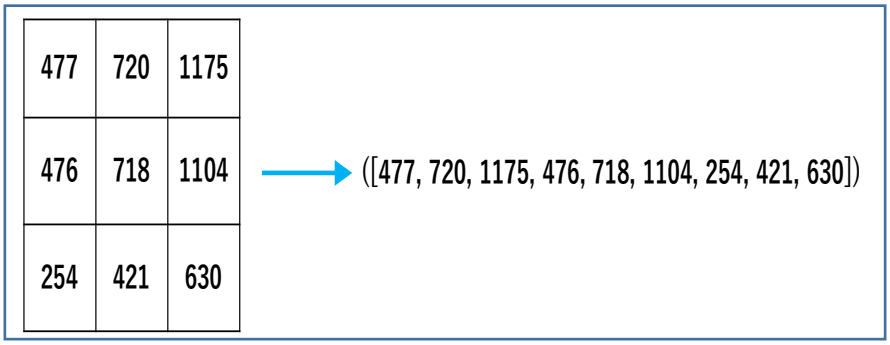


図 5 -6 多次元情報の一次元への平坦化

全結合層で 1 次元にされた情報と重みを用いて，出力層では，図 5 -7 のようにソフトマックス関数[注 57]により，画像が 0～9 のどれに該当するかそれぞれ確率を計算し，最も高確率だった文字が分類結果となる．

注57 出力結果を 3 つ以上に分類する多値クラス分類において広く使用されている．シグモイド関数を拡張して，出力分類を 3 つ以上に対応させたのがソフトマックス関数である．

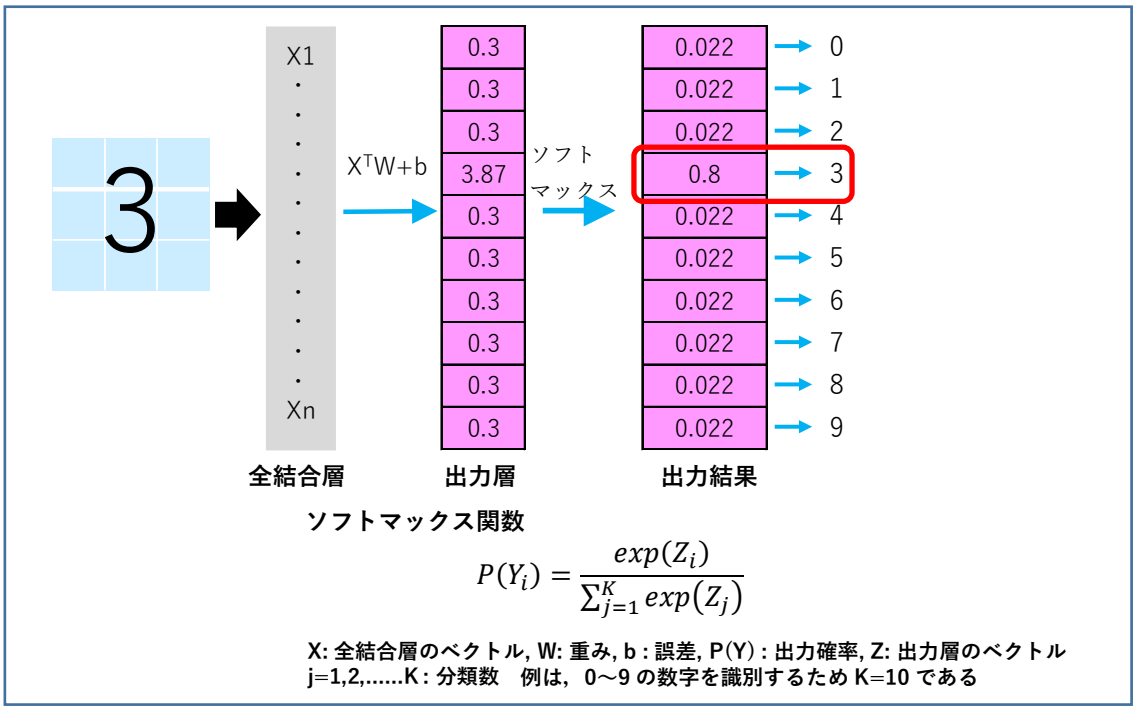


図 5 -7 出力層における結果の出力

◆　画像診断に用いられる学習済みモデル

2019 年 3 月 28 日に理化学研究所と東北大学の共同研究グループから，眼底検査装置からのマルチモダリティ画像情報を用いた緑内障を自動診断できる機械学習モデル構築が発表された[注 58]（第 5 章第 5 節第 1 項参照）．このモデルには転移学習が用いられ，VGG19 という CNN 構造を有する学習済みモデルが採用されている[68]．VGG は，動物，花，楽器，文房具など 1000 個のカテゴリの分類について 100 万枚を超える画像にて学習されており，医療分野における幅広い領域で画像診断への応用も試みられている[69,70,71]．VGG は，2014 年の ILSVRC に参加したチーム名で，オックスフォード大学の Visual Geometry Group に由来している[72]．畳み込み層，全結合層及び出力層を含めて，全部で 16 層若しくは 19 層からなり，層の深さに応じて，「VGG16」や「VGG19」と呼ばれている．VGG19 のアーキテクチャとモデルサマリーを図 5 -8 及び図 5 -9 に示す．入力層における 3 次元配列は 224×224×3 であるが，畳み込み層とプーリング層を繰り返すことで，7×7×512 となり，画像データにおける数値情報が縮約され，画像の特徴量を抽出している．なお，VGG16 は，VGG19 にある Conv3_4, Conv4_4 及び Conv5_4 の 3 つの層が含まれていないが，基本コンセプトは同一である．

注58 理研と東北大など，マルチモダリティ情報を用いて緑内障を自動診断できる機械学習モデルを構築. 日本経済新聞. 2019-03-28. https://www.nikkei.com/article/DGXLRSP506325_Y9A320C1000000/

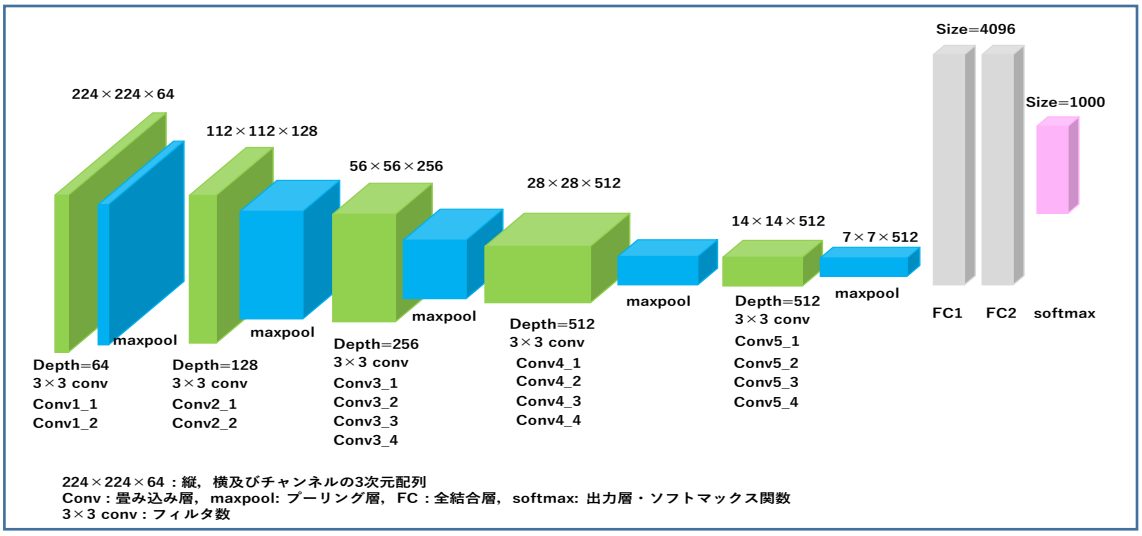


図 5 -8 VGG19 アーキテクチャ[73]

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000

Total params: 143,667,240
Trainable params: 143,667,240
Non-trainable params: 0

図 5 -9 VGG19 モデルサマリー

VGG19 を利用してエレキギター(Image1), ブルドック(Image2)及び皮膚がん(Image3)の画像認識を行った結果を図 5 -10 に示す．エレキギターの画像は，画像認識結果の中で最も高い確率が Electric guitar であった．ブルドックの画像は，画像認識結果の中で最も高い確率が French bulldog であった．一方，皮膚がんの画像は，画像認識結果の中で Flatworm（扁虫）や Isopod（ワラジムシ）と認識された．エレキギターやブルドックは，正しく認識されたが, 皮膚がんについては, 正しく認識されなかった. この理由は, VGG19 が ImageNet という日常の生活の中で遭遇する楽器や動物といった画像で事前に学習されており，皮膚がんは分類クラスに含まれていないためである．

皮膚がんかどうかを判定させるためには，VGG19 を良性のほくろ又は悪性の皮膚がんの画像とこれらを識別する教師ラベルデータを用いた転移学習やファインチューニングといった手法が必要となる．緑内障を自動診断できる機械学習モデルの構築も転移学習やファインチューニングといったアプローチに基づいている．

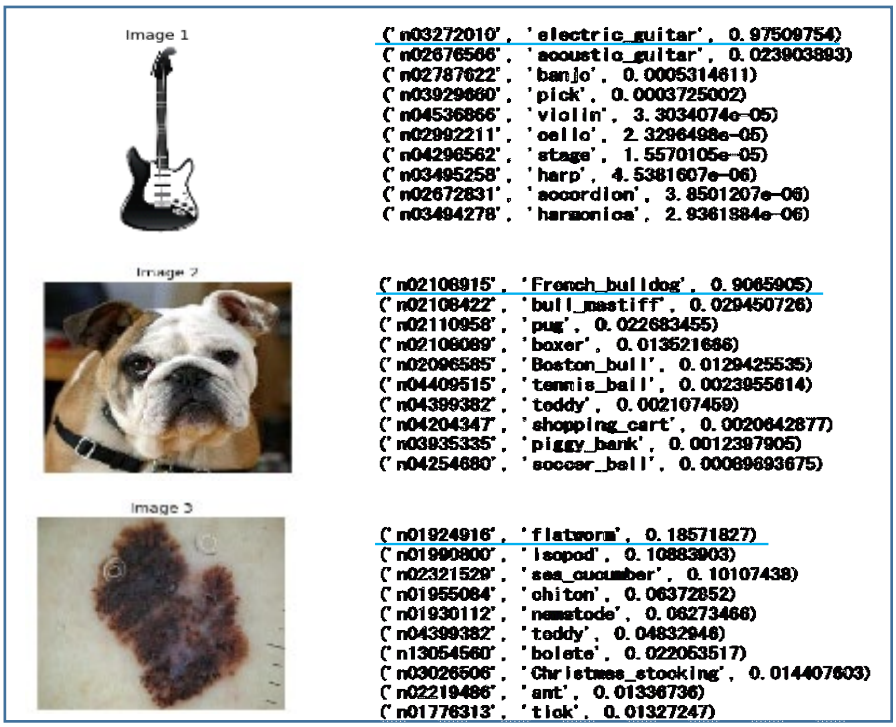
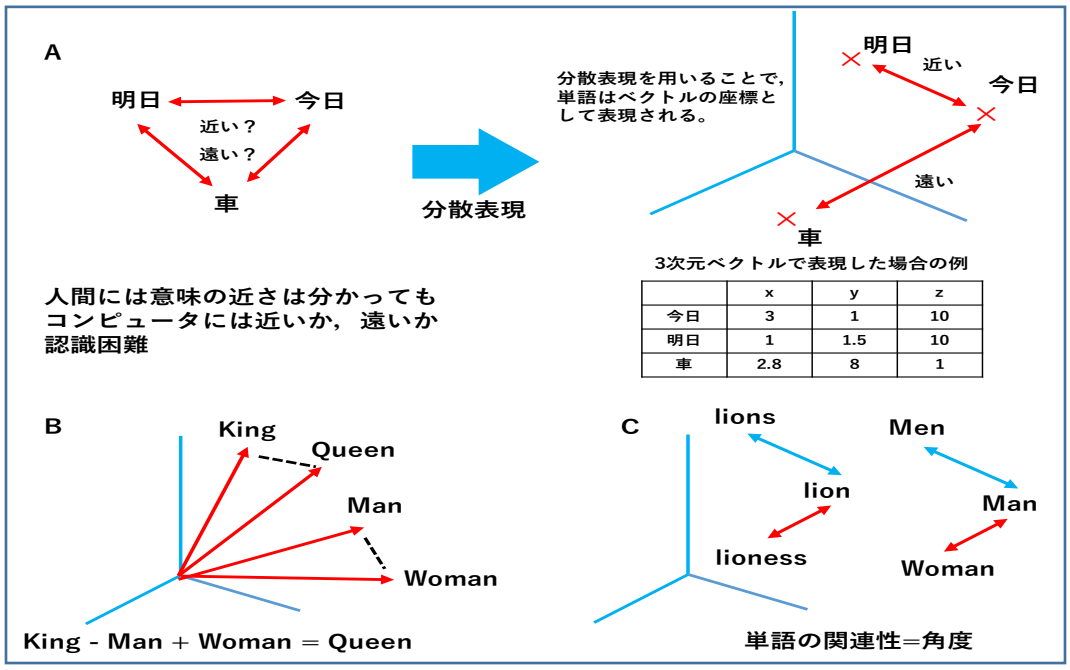


図 5 -10 VGG19 による画像認識[注 59]

注59 エレキギター（Image1），ブルドック（Image2）は，PublicDomainPictures.net（https://www.publicdomainpictures.net/en/)から CC0 Public Domain ライセンスで公開されている画像を用いた．皮膚がん（Image3）は，Kaggle にて公開されている良性のほくろ又は悪性の皮膚がんの画像（https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign)を利用した．詳細は，第 4 節を参照.

B） 自然言語処理に用いられる学習済みモデル

人間が日常的に使用している言語や言葉をコンピュータに処理させる技術を自然言語処理（Natural Language Processing, NLP）と呼ぶ．画像認識分野では，画素情報などの数値を利用して計算を行うことが可能であるが，人間がコミュニケーションなどで使用している言語や言葉は，数値ではないため，そのままでは計算することができない．そこで，分散表現という図５-11-A のように単語や複合語などの自然言語の構成要素をベクトル空間に埋め込む方法がある．ベクトル空間により，点と点との距離，角度を定義することができ，加算といった演算も可能となる．この単語埋め込みベクトルを利用することで，図５-11-B のように"King - Man + Woman = Queen"といったように単語の意味を捉えることや図５-11-C のように単語間の関連性など単語を使った意味的な類推をベクトル演算で実現することができる[53,74,75,76]．



	x	y	z
今日	3	1	10
明日	1	1.5	10
車	2.8	8	1

図５-11 分散表現による単語埋め込みベクトル[76]

こうした手法は，品詞のタグ付け，文書分類，評判分析，機械翻訳といった自然言語処理に応用されている[77]．単語埋め込みの代表的な手法として 2013 年に Google からリリースされた Word2vec モデル[53]がある．図５-12 のように，Continuous Bag of Words（CBOW）と Skip-Gram という２つのバージョンがある．CBOW は，文脈（周囲の単語）からある単語を予測するモデルであり，Skip-Gram は，ある単語が与えられたときの文脈（周囲の単語）を予測するモデルである．そのほかの単語埋め込みとして GloVe, fastText 及び sent2vec といったモデルがある[54, 55, 56]．

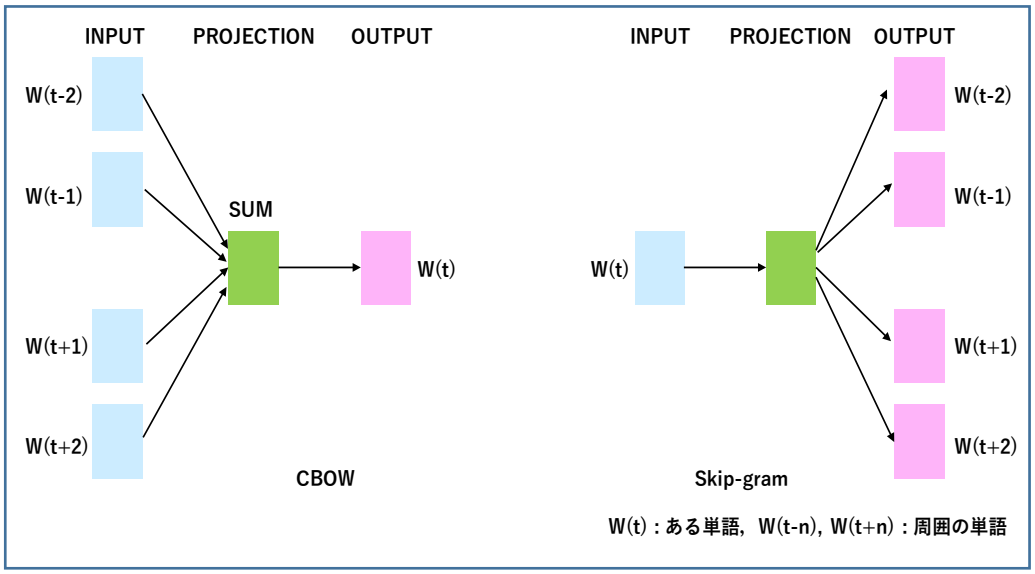


図 5 -12 Word2vec モデル[53]

Word2vec などで用いられている単語埋め込みベクトルは，Wikipedia などの大量のコーパス（文書データ）に基づいて構築され，数百万の語彙間の関係について網羅的な表現を獲得している．従って，日常で遭遇する言語が対象となっている．こうした手法は，専門用語を含むコーパスを用意すれば，特定の領域への応用も可能である．表 5 -1 のように，医療分野においても PubMed などから疾患，治療といった医療分野に関する大量の情報を用いて学習した単語埋め込みモデルや単語埋め込みベクトルを利用した学習済みの言語モデルが公開されている．

表 5 -1 PubMed などで事前学習した言語モデル

公開先	ライセンス	学習済み言語モデル
BioASQ http://bioasq.org/	Public Domain	PubMed Abstracts から生物医学論文の 10,876,004 コーパスで事前に学習したモデル．Word2vec の Skip-Gram が利用されている． ・BioASQword2vec http://bioasq.lip6.fr/info/BioASQword2vec/
RaRe Technologies https://github.com/RaRe-Technologies/	CCBY	MEDLINE/PubMed 2018 の 2,700 万件の生物医学論文からのテキストで事前トレーニングされた単語埋め込みモデル．次元数に応じて 2 つのバージョンが存在する．Word2vec の Skip-Gram が利用されており，BioASQword2vec の改良版である． ・pubmed2018_w2v_200D ・pubmed2018_w2v_400D

公開先	ライセンス	学習済み言語モデル
NCBI BioNLP Research Group https://github.com/ncbi-nlp/	Public Domain	PubMed 抄録及び MIMIC-III Clinical notes で事前に学習した生物医学的な単語と文章の埋め込みモデル．ワードベクトルとハイパーパラメータを含むモデルが利用可能である．単語の埋め込みは fastText モデルが利用されており，Skip-Gram による Word2vec モデルに比べ埋め込みの精度が向上している． ・BioWordVec vector ・BioWordVec model https://www.nature.com/articles/s41597-019-0055-0/
		PubMed 抄録及び MIMIC-III Clinical notes で事前に学習した生物医学的な単語と文章の埋め込みモデル．単語の埋め込みは sent2vec モデルが利用されている． ・BioSentVec https://arxiv.org/abs/1810.09302/
		PubMed に基づく生物医学的概念埋め込みモデル．遺伝子，突然変異，化学物質，病気と細胞株をカバーし，訓練された埋め込みは 400,000 以上の概念を含む．利用した単語埋め込みモデル（CBOW, Skip-Gram, Glove, fastText）に基づき 4 つのバージョンが存在する． ・BioConceptVec cbow ・BioConceptVec skip-gram ・BioConceptVec glove ・BioConceptVec fastText https://github.com/ncbi-nlp/BioConceptVec/
		PubMed 抄録及び MIMIC-III Clinical notes で事前に学習した言語モデル．Attention メカニズムに基づくニューラルネットワーク言語モデルである BERT が利用されている．次元数に応じて，それぞれ 2 つバージョンが存在する． ・NCBI_BERT-Base, Uncased, PubMed ・NCBI_BERT-Large, Uncased, PubMed ・NCBI_BERT-Base, Uncased, PubMed+MIMIC-III ・NCBI_BERT-Large, Uncased, PubMed+MIMIC-III https://arxiv.org/abs/1906.05474/
Biomedical natural language processing http://bio.nlplab.org/	CCBY	MEDLINE/ PMC/ Wikipedia で事前トレーニングされた単語埋め込みモデル．ソースデータの組み合わせにより 4 つのバージョンが存在する． ・PMC-w2v ・PubMed-and-PMC-w2v ・PubMed-w2v ・wikipedia-pubmed-and-PMC-w2v http://bio.nlplab.org/pdf/pyysalo13literature.pdf/

PubMed にて事前学習した PubMed-w2v モデル（公開先：Biomedical natural language processing https://bio.nlplab.org/）を用いて，医療分野における類似単語の抽出を行った結果を表 5 -2 に示す.「diabetes（糖尿病）」と類似性の高い単語として,「T2DM」や「NIDDM」といった単語が抽出され，「endometriosis（子宮内膜症）」と類似性の高い単語として，「adenomyosis（子宮腺筋症）」や「fibroids（子宮筋腫）」といった単語が抽出された.

表 5 -2 類似性の高い単語の抽出

【糖尿病（diabetes）と類似性の高い単語】	【子宮内膜症（endometriosis）と類似性の高い単語】
('T2DM',0.8334260582923889)	('adenomyos is', 0 .8583588600158691)
('T1DM', 0.7870343923568726)	('endometrioma', 0.7309424877166748)
('DM-2', 0.7731763124465942)	('myomas', 0.7142377495765686)
('NIDDM', 0.7693204879760742)	('endometriomas', 0.7071224451065063)
('DM2', 0.7682093381881714)	('leiomyomata', 0.7019719481468201)
('mellitus', 0.7621428966522217)	('hydrosalpinx', 0.6970451474189758)
('non-i nsulin- dependent', 0.7576709985733032)	('fibroids', 0.6960325241088867)
('prediabetes', 0.7512259483337402)	('deep-infiltrating', 0.6811408996582031)
('PIDDM', 0.7462856769561768)	('menor rhag ia', 0 .6796872615814209)
('T2D', 0.7360433340072632)	('myoma', 0.679 3314218521118)

また，単語埋め込みベクトルを用いた演算により意味的な類推を行った結果を表 5 -3 に示す.「prostatic（前立腺の）」,「hyperplasia（過形成）」及び「LUTS（下部尿路機能障害によって起こる排尿・蓄尿に関する症状の総称）」を足し合わせて「cancer（癌）」を引いたところ，「BPH（前立腺肥大症）」や「hyperplasia/benign（良性/過形成）」となった．前立腺肥大症診療ガイドラインにおける「前立腺の良性過形成による下部尿路機能障害を呈する疾患で，通常は前立腺腫大と下部尿路閉塞を示唆する下部尿路症状を伴う」という定義[78]とも大きな差異はなかった．このように，文字の認識のみならず医学的な意味も捉えており，特定の領域の語彙への応用も可能である.

表 5 -3 演算による単語を使った意味的な類推

【prostatic + hyperplasia + LUTS – cancer 】
('BPH', 0.7 107576727867126)
('symptoms/benign'. 0.6783807277679443)
('hyperp las ia/ benign'. 0.6574642658233643)
('prostatatic '. 0.6505674123764038)
('/benign'. 0.6230175495147705)
('hyperplasia/lower'. 0.6108564138412476)
('Postatrophic ', 0.6068457365036011)
('hyperplasia- related', 0.6054283380508423)
('papillary /nodular', 0.5903332829475403)
('hist iocyt ic/mesothelial ', 0.5 771628022193909)

第 2 項 日本語を含む多言語に対応した自然言語処理ライブラリ

日本語は英語のようなスペースで単語が区切られておらず，文を構成する単語の境界は明確でないため，文を構成する単語の境界を明らかにする必要がある．この処理は形態素解析と呼ばれ，単語分割に加えて，名詞や形容詞といった品詞のタグ付けも行う[79]．この品詞情報を用いることで，文書中から名詞だけを取り出し，キーワードを抽出することも可能である．日本語に対応した代表的な形態素解析ソフトウエアとして，MeCab（http://taku910.github.io/mecab/）などがある．一方で，Universal Dependencies（UD）（https://universaldependencies.org/）という多言語間で共通のアノテーション方式を用いた言語横断的な学習による言語間での定量的な比較といった単語依存構造解析が試みられている．UD は，17 種類の品詞タグセットに基づく，多言語間で共通した構文構造アノテーションを用いて，図 5 -13 のような単語の相互依存関係だけを記述し，句構造を考慮しないことにより，表現が単純化され，言語資源の作成の省力化ができるのみならず，くだけた表現や特殊な言い回しに関しても頑健な表現となるといった利点があると言われている[80,81]．

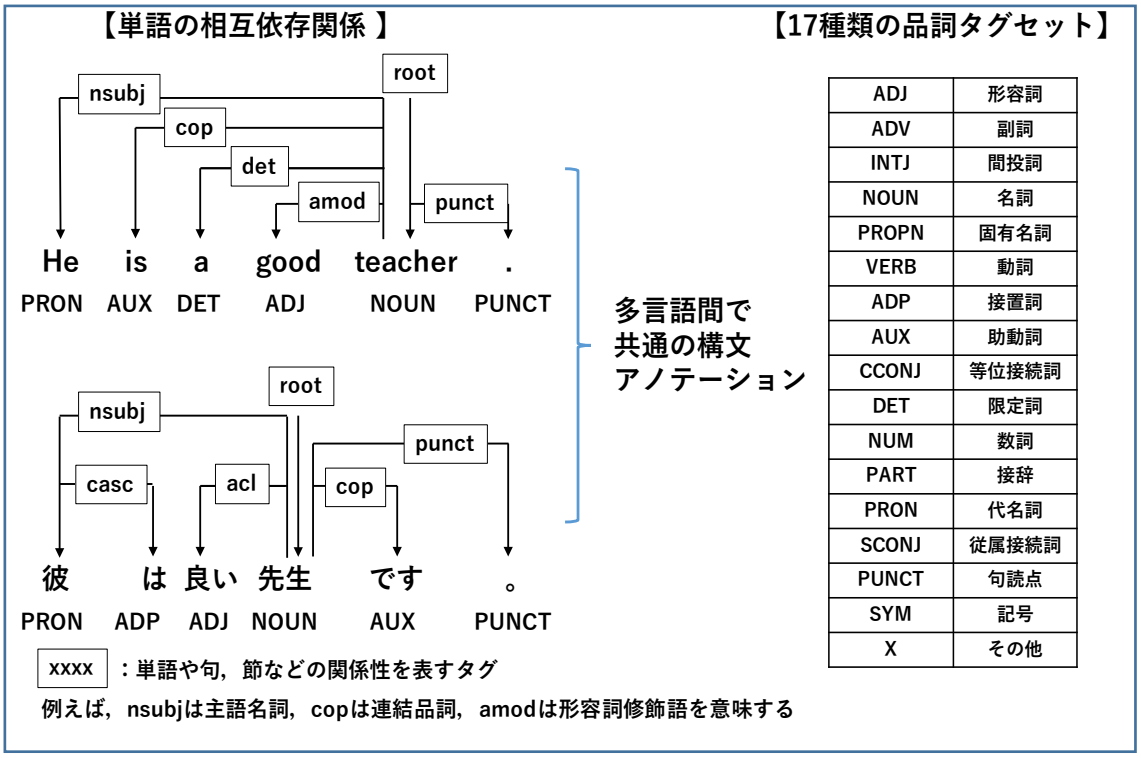


図 5 -13 Universal Dependencies における単語の相互依存関係[80, 81]

こうしたなか，2019 年 4 月 2 日に株式会社リクルートから「リクルートの AI 研究機関，国立国語研究所との共同研究成果を用いた日本語の自然言語処理ライブラリ「GiNZA」を公開」が発表された[82]．この「GiNZA（https://megagonlabs.github.io/ginza/）」

は，UD に基づくオープンソースな日本語を含む多言語にも対応した自然言語処理ライブラリであり，MIT ライセンスの元で公開されている．GiNZA は，日本語版 UD に基づいた高精度な依存構造解析研究の結果を組み込んだ学習済みモデルであり，同じく UD に基づいて設計された自然言語ライブラリである「spaCy（https://github.com/explosion/spaCy)」をフレームワークとして利用している[82,83]．したがって，複数の欧米言語と日本語の言語ソースを切り替えて使用することができ，複数の言語を単一のライブラリにて実装することが可能である．

MeCab 及び GiNZA にて形態素解析を行った結果を表 5 -4 及び表 5 -5 に示す．形態素解析は，PDF 形式の日本語版及び英語版のヘルシンキ宣言[84]をテキスト形式に変換した文章を対象とした．まず，以下の短い平文について，形態素解析を行った．

日本語：ヘルシンキ宣言　人間を対象とする医学研究の倫理的原則
英語：DECLARATION OF HELSINKI Ethical Principles for Medical Research Involving Human Subjects

MeCab においては，日本語では名詞や動詞といった品詞の分割が適切に行われたが，英語による学習を行っていないため，英語では，全て名詞と判定された．GiNZA においては，言語ソースを切り替えることで，UD よる同一のアノテーションルールに基づく日本語及び英語の両言語において形態素解析が可能であった．

表 5 -4 MeCab による形態素解析結果（左：日本語，右：英語）

ﾍﾙｼﾝｷ名詞,固有名詞,地域一般, *,*, ﾍﾙｼﾝｷ,ﾍﾙｼﾝｷ,ﾍﾙｼﾝｷ	DECLARATION 名詞,固有名詞,組織*,*,*,*
宣言 名蒴,ｻ変接続,*,*,* ,*,宣言,ｾﾝｹﾞﾝ,ｾﾝｹﾞﾝ	0F 名詞,一般*,屯*,*,*
人間 名詞,一般,*,*,*,*,人間,ﾆﾝｹﾝ,ﾆﾝｹﾞﾝ	HELSINKI 名詞,一般,*,*,*,*,*
を 助詞,格助詞,一般*,*,*,を,ヲ,ヲ	Ethical 名詞,一般*,*,*,*,*
対象 名詞,一般,*,*,*,*,対象,ﾀｲｼｮｳ,ﾀｲｼｮｰ	Principles 名飼,一般*,屯*,*,*
と 助詞,格助詞,一般,*,* ,*, と,ト,ト	for 名詞,一般*,屯*,*,*
する 動詞,自立*,*,ｻ変･ｽﾙ,基本形,する,ｽﾙ,ｽﾙ	Medical 名詞,一般,*,*,*,*,*
医学 名詞,一般,*,*,*,*,医学,ｲｶﾞｸ,ｲｶﾞｸ	Research 名詞一般*,*,*,*,*
研究 名詞,ｻ変接紐,*,*,*,* ,研究,ｹﾝｷｭｳ,ｹﾝｷｭｰ	Involving 名蒴一般*,*,*,*,*
の 助詞,連体化,*,*,*,*,の,ﾉ,ﾉ	Human 名詞一般*,*,*,*,*
倫理 名詞,一般, 丸* ,*,*,倫理,ﾘﾝﾘ,ﾘﾝﾘ	Subjects 名詞,固有名詞,組織,*,*,*,*
肘 名詞,接尾,形容動詞語幹,*,*,*,的,ﾃｷ,ﾃｷ	EOS
原則 名詞,一般,*,*,*,*, 原則,ｹﾞﾝｿｸ,ｹﾞﾝｿｸ	
EOS	

表 5 -5 GiNZA による形態素解析結果（左：日本語，右：英語）

0 ヘルシンキ PROPN 名詞-固有名詞-地名-一般 compound 2	0 DECLARATION PROPN NNP ROOT 0
1 宣言 NOUN 名詞-普通名詞-サ変可能 compound 2	1 OF ADP IN prep 0
2 人間 NOUN 名詞-普通名詞-一般 obj 6	2 HELSINKI PROPN NNP compound 4
3 を ADP 助詞-格助詞 case 2	3 Ethical PROPN NNP compound 4
4 対象 NOUN 名詞-普通名詞-一般 nmod 6	4 Principles PROPN NNPS pobj 1
5 と ADP 助詞-格助詞 case 4	5 for ADP IN prep 4
6 する AUX 動詞-非自立可能 acl 8	6 Medical PROPN NNP compound 7
7 医学 NOUN 名詞-普通名詞-一般 compound 8	7 Research PROPN NNP pobj 5
8 研究 NOUN 名詞-普通名詞-サ変可能 nmod 11	8 Involving VERB VBG acl 0
9 の ADP 助詞-格助詞 case 8	9 Human PROPN NNP compound 10
10 倫理的 ADJ 名詞-普通名詞-形状詞可能 compound 11	10 Subjects NOUN NNS dobj 8
11 原則 NOUN 名詞-普通名詞-副詞可能 ROOT 11	EOS
EOS	

また，GiNZA にて形態素解析を行い，日本語及び英語のヘルシンキ宣言から名詞（固有名詞及び普通名詞）を抽出して，頻出上位 10 単語の集計を行った．

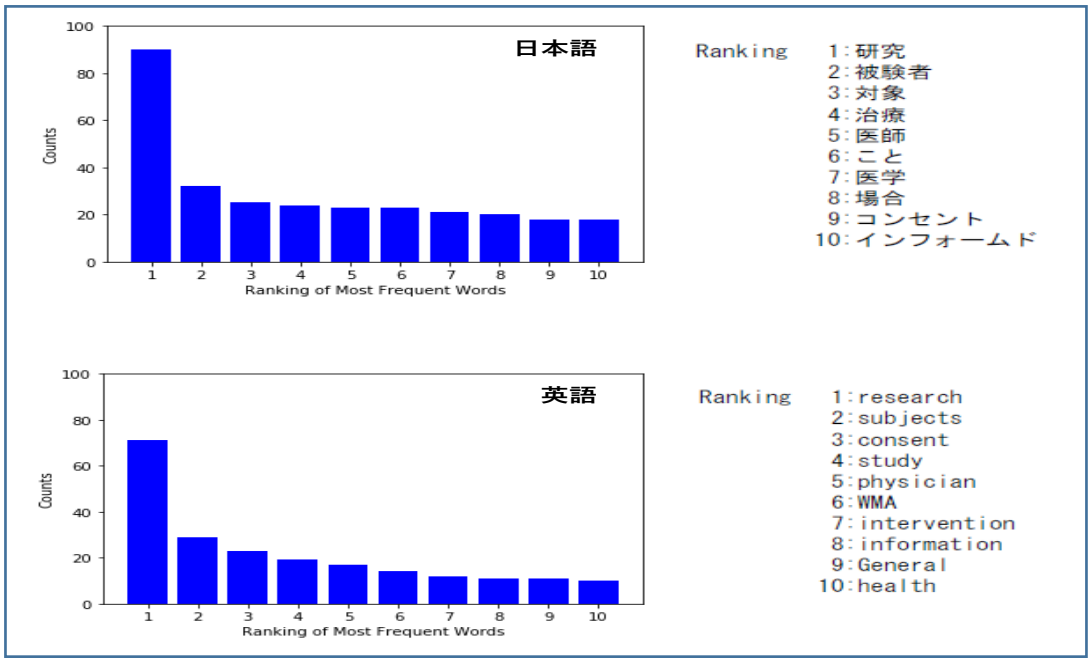


図 5 -14 ヘルシンキ宣言における頻出上位 10 単語の集計

図 5 -14 に示したように，日本語においては，「研究」，「被験者」，「治療」，「インフォームド」及び「コンセント」といったキーワードが抽出された．英語においても「research」，「subjects」，「consent」及び「physician」といったキーワードが抽出された．日本語及び英語の両言語において，頻出上位 10 単語は，ヘルシンキ宣言の内容から考えられる妥当な単語であった．

第 3 節 転移学習

第 1 項 転移学習の概要

学習済みモデルを利用して新しい識別や予測に活用することが転移学習である[85]．通常の機械学習的手法では，それぞれ固有の領域（ドメイン）の教師付き学習データに基づき一からモデルを構築する必要がある．転移学習は，識別，予測などの問題を効果的かつ，効率的に解くために別の関連した問題（ソースドメイン）のデータや学習結果を利用することにより，関連しているが異なる部分を有するデータから，目的の問題（ターゲットドメイン）に利用できる情報や知識だけを取り込んで，より高い予測精度を得る手法である[45, 46]．

深層学習による識別や予測を行う場合に必要なのは，あくまでも最適化されたネットワークの重み（特徴量）[85]であり，これが知識に該当する．既に，大量のデータにて学習したネットワークの重みがあるため，図 5 -15 に示したように，最初からモデルを構築する場合に比べて学習時間を短縮でき，より少ないデータでも精度の高いモデル構築が行えるのが転移学習の利点である．

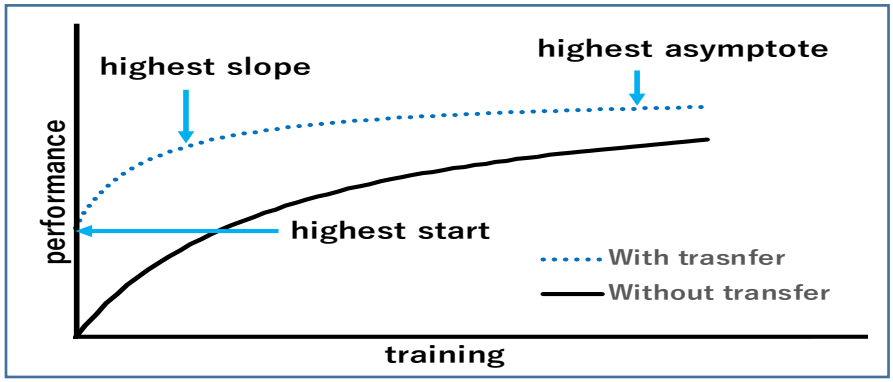


図 5 -15 転移学習の利点[86]

転移学習を考える上で，ドメイン（観察データ，予測ラベル，識別や予測の確率を包括したデータの集まり）とタスク（分類や予測に関する作業や処理）といった概念への理解が必要となる．機械学習においてドメインは学習データ，背景知識，適用するビジネスなどを包括する分野の意味合いであることが多いが，転移学習においてドメイン:D は，特徴ベクトル空間:χ 及び周辺確率 P(X)により定義でき，D={χ,P(X)}と表現される．タスクは機械学習により得られたモデルの働きであるが，転移学習におけるタスク:T は，図 5 -16 に示したように，ラベル空間:y 及び目的関数:η と定義でき，T={y，η}と表現される．この η は，観測データが X，予測ラベルが Y の時の条件付き確率分布 P(Y|X)となる[46]．

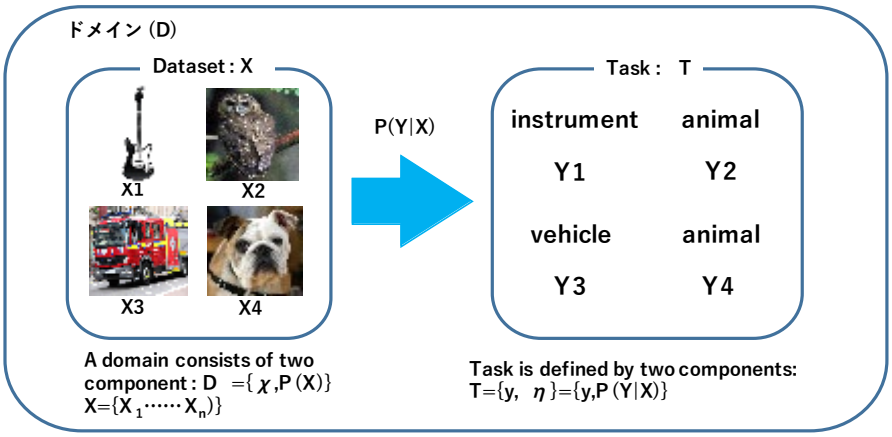


図 5 -16 ドメインとタスクの表現[注 60]

図 5 -17 に示したように，画像は異なるが同じフクロウ，同じブルドックといったソースドメインとターゲットドメインのデータは類似しており，ソースタスクとターゲットタスクが互いに異なる場合は転移学習が可能なケースである．

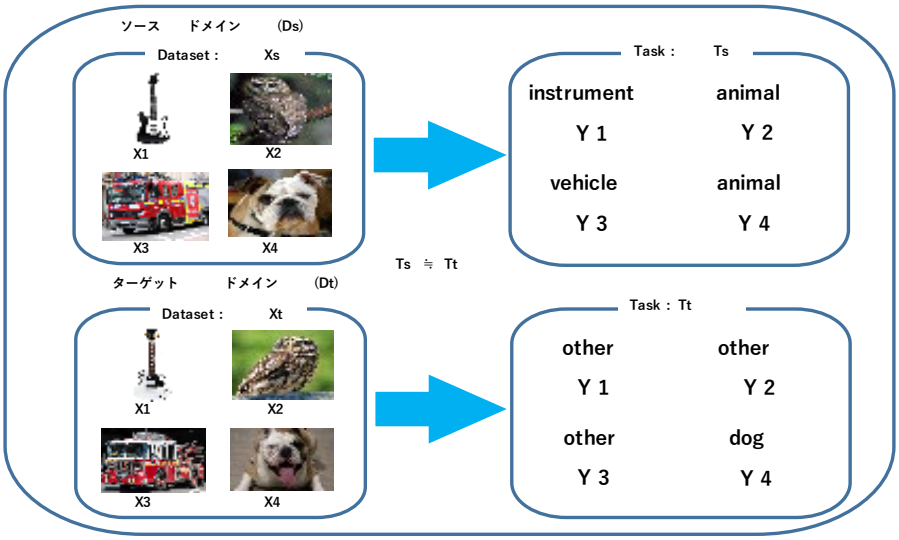


図 5 -17 転移学習が可能なケース 1[注 60]

図 5 -18 に示したように，ソースタスクとターゲットタスクは同じであるが，対応するドメインデータが異なる場合（対象物自体は関連している，例：同じ鳥でもフクロウとインコ，同じ犬でもブルドックとシェットランドシープドッグ）も転移学習が可能なケースである．

注60 転移学習が可能なケースの説明で用いた画像は，PublicDomainPictures.net（https://www.publicdomainpictures.net/en/）から CC0 Public Domain ライセンスで公開されている画像を用いた．（転移学習が可能なケース 3 のターゲットドメイン画像（皮膚がんの画像）を除く）．詳細は，第 4 節を参照．

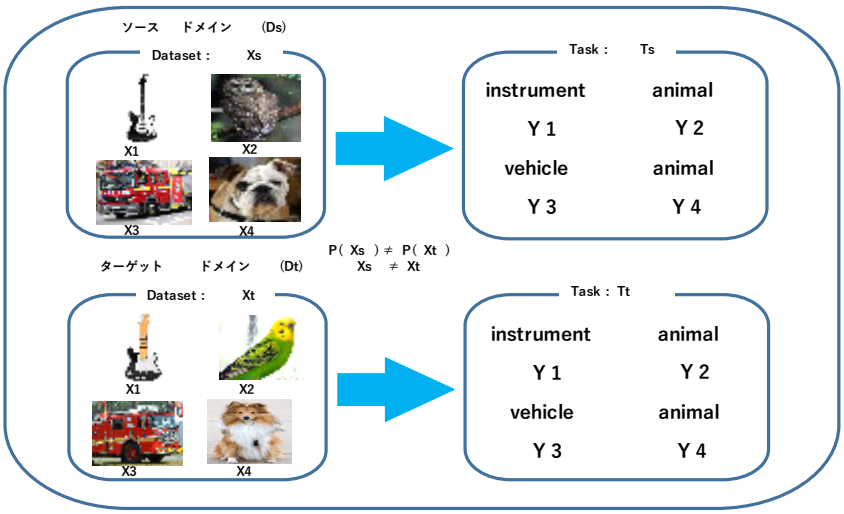


図 5 -18 転移学習が可能なケース 2 注 60

図 5 -19 に示したように，ソースタスクとターゲットタスクが互いに異なり，対応するドメインのデータも異なるが，同じ画像認識という面では，第 2 節で紹介した畳み込み層が保持している視覚的特徴を用いることで，転移学習が可能となるケースである．

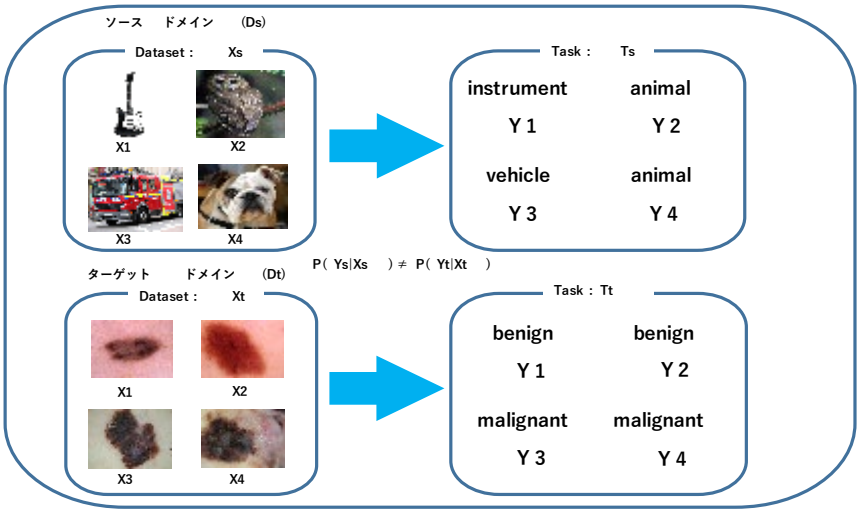


図 5 -19 転移学習が可能なケース 3 注 61

こういった例示を含めて様々な転移学習の可能なケースが考えられるが，実際に転移学習を行うにあたり，考えるべき要素として以下の 3 つが示されている[46]．

- 何を転移するのか（what to transfer）
  転移学習を行うプロセス全体で最初の最も重要なステップに該当し，ターゲットタスク

注61 ターゲットドメインの皮膚がんの画像は，Kaggle にて公開されている良性のほくろ又は悪性の皮膚がんの画像(https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign/)を利用した．

の精度を改善させるために，ソースからターゲットへ転移させる部分を探し，ソースとターゲットの間の共通点やソース特異的な部分を特定する必要があること

- どのように転移させるのか（How to transfer）
  既存方法の変更や異なる技術を用いて，ドメイン・タスクの間での知識の転送を行うための方法を探し続ける必要があること
- いつ転移するのか（When to transfer）
  知識の転移を行うことで改善よりむしろ改悪させる場合があるので，ターゲットタスクの性能を向上させ，低下させないために転移させるか否かのタイミングへの配慮が必要となること

更に，ソースドメイン，ターゲットドメイン，ソースタスク及びターゲットタスクの特性並びに教師付きラベルの有無によって様々なシナリオに基づく転移学習のアプローチも提案されている[46]．教師付きラベルの有無によって表 5 -6 に整理できる．

表 5 -6 転移学習の教師付きラベルの有無による分類の整理

		ターゲットの教師付きラベル	
		あり	なし
ソースの教師付きラベル	あり	帰納的転移学習（Inductive Transfer Learning）	トランスダクティブ転移学習（Transductive Transfer Learning）
	なし		教師なし転移学習（Unsupervised Transfer Learning）

一方で，学習データ以外に最初から持っている仮説や仮定は帰納的バイアスと呼ばれている[87]．転移学習では，図 5 -20 に示したように，ソースタスクから得た知識と帰納的バイアスを利用してターゲットタスクの調節や選択を行うことで，異なるタスクの予測が可能と報告されている[86]．

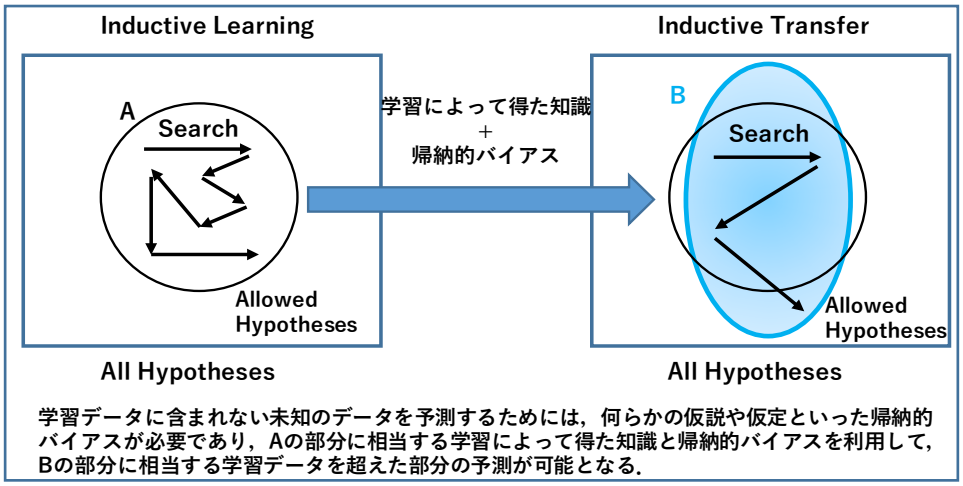


図 5 -20 帰納的学習と帰納的転移[86]

また，ある予測問題について手法 A が手法 B よりも性能が良かったとしても手法 B の方が手法 A よりも良くなる別の予測問題が必ず存在するという「ノーフリーランチ定理」が知られている[87]．すなわち，「あらゆる問題で性能の良い機械学習モデルは理論上不可能であり，あるモデルが他のモデルより性能が良いのは，解こうとしている特定の問題に対して専門化または特化されている場合のみである」ということを意味している[88]．実際に転移学習を活用するにあたり，ソースとターゲット及び教師付きラベルの有無といった特性やこうした定理にも留意することが望まれる．

次に，学習済みモデルを用いた転移学習に関連する代表的な手法を紹介する．なお，ファインチューニングについては，第 4 節で実装事例も交えて別途に紹介する．

第 2 項 転移学習に関連する代表的な手法

A） 特徴抽出（Feature Extraction）

学習済みのニューラルネットワークを特徴抽出器として使用する方法である[89]．ターゲットタスクの画像を学習済みの畳み込みニューラルネットワーク（CNN）に入力して得られた中間層の出力を特徴ベクトルとして利用する．これは，ネットワークのすべての層は画像からある種の特徴を抽出する役割を持つという性質に基づく．こうして取り出した特徴量は，サポートベクトルマシンやランダムフォレストといった機械学習モデルに入力して分類を行う．この手法は，ターゲットタスクの学習サンプルが大量に用意できない場合であっても学習済みの重みを流用することで，高い予測性能の実現が可能となる[89]．実際に，理化学研究所と東北大学の共同研究グループが発表した緑内障を自動診断できる機械学習モデル構築の際にもこの手法が用いられている．例えば，CNN による特徴抽出を行う場合は，畳み込み層でターゲットタスクのデータを処理し，その出力ベクトルに基づいて分類器を学習させ，通常，全結合層を選んで特徴ベクトルとして用いる[89]．これは，全結合層から抽出した特徴には視覚的な情報が除かれており，畳み込み層には汎用的な視覚的特徴が保持されているためである．つまり，入力層から出力層へ進むに連れて画像に対する汎用的な視覚的特徴は減り，これとは逆にデータに依存した画像クラス分類の特徴に関する情報が増えることになる[90]．したがって，図 5 -21 に示したように，汎用的な視覚的特徴を利用するために入力に近い部分の重みを固定し，出力に近い部分だけを学習させることで新しいタスクへの適用が可能となる．

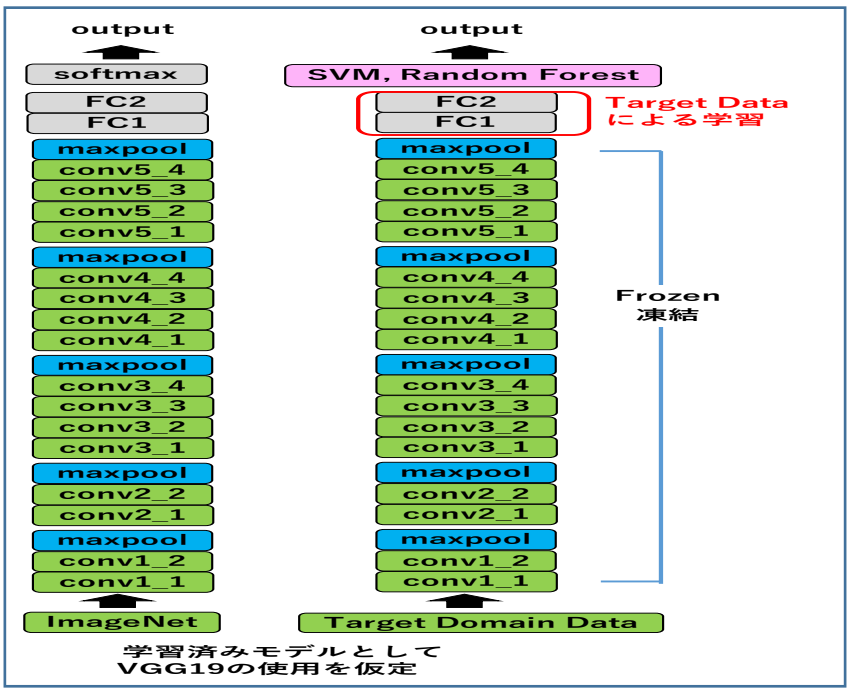


図 5 -21 学習済みモデルを利用した特徴抽出

この重みを固定することはフローズン（Frozen，凍結）と呼ばれている．「図 5 -17 転移学習が可能なケース 1」のような場合への適用が考えられ，ソースタスクとターゲットタスクのデータの内容が著しく異なる場合は，全結合層と畳み込み層の両方を学習させるといったファインチューニングのプロセスが必要となる．

B） ドメイン適応（Domain Adaptation）

ドメイン適応は,「図 5 -22 ドメインシフト」のような場合に有効な手段である．例えば，フクロウとイヌコやブルドックとシェットランドシープドッグといったように種類は異なるが，鳥や犬又は動物として識別したい場合などが挙げられる．こういったケースでは，図 5 -22 のように，対象物自体は関連性があるが，ソースドメインとターゲットドメインが異なると分布や特徴空間が異なるドメインの変化（Domain Shift）を生じる．目的とするタスクは同一であるが，異なるドメインでデータが収集された場合，それらのデータ同士で知識を転移させ，異なる分布・特徴空間を近づけるように調整していくことが必要になる．このことをドメイン適応という[91]．

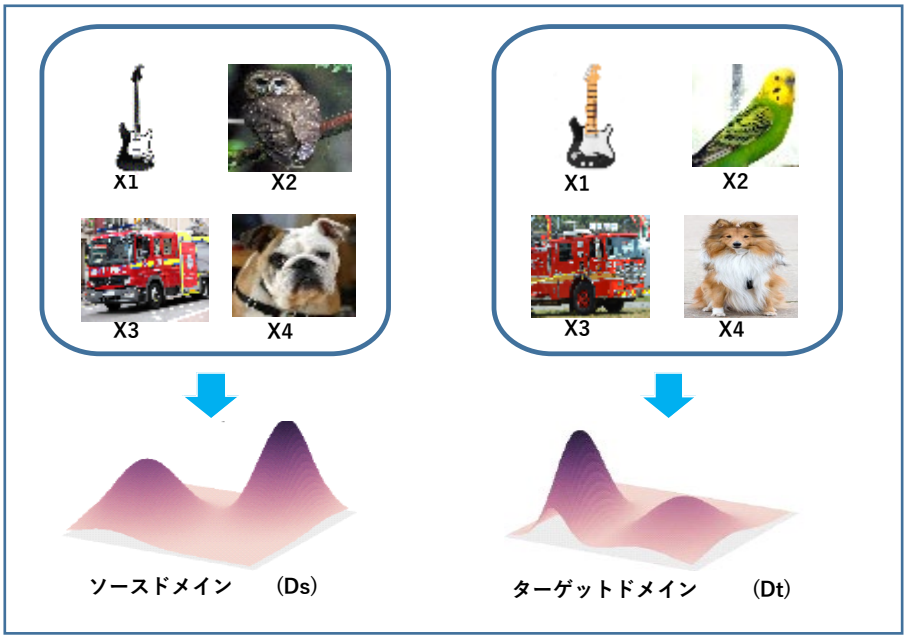


図 5 -22 ドメインシフト[注 60]

ドメイン適応は，One-step ドメイン適応と Multi-step ドメイン適応といった 2 つの学習プロセスがあり，ターゲットドメインにおける教師付きラベルの有無により，教師ありドメイン適応（Supervised Domain Adaptation），半教師ありドメイン適応（Semi-supervised Domain Adaptation）及び教師なしドメイン適応（Unsupervised Domain Adaptation）に分類される[91]．また，学習プロセス及びターゲットタスクにおける教師付きラベルの有無に応じた種々の手法が提案されている[92,93,94]．

C） ゼロショット学習（Zero-Shot Learning）

深層学習の自動運転への応用を考えた場合，道路標識，障害物及び歩行者といった認識を行わせるために，膨大な画像データと共に，これらに付随した多種類の教師付きラベルが必要となる．しかし，現実的には数え切れない量の物体があり，全ての教師付きラベルを施した画像データを集めることが不可能である．このようなケースでは，学習データに存在しない画像を分類させる必要がある[95]．こういった未知クラス分類の推定に用いられるのがゼロショット学習である．

ゼロショット学習のコンセプトは，画像から得られる特徴とは別の特徴から未知の画像のラベルを推定することである．具体的には，図 5 -23 のように，前項で紹介した単語の埋め込みベクトルである分散表現の特性を利用して，未知データのクラス分類の推定を行う[95,96]．このように関連した知識を流用するため，転移学習の一つと考えられる．

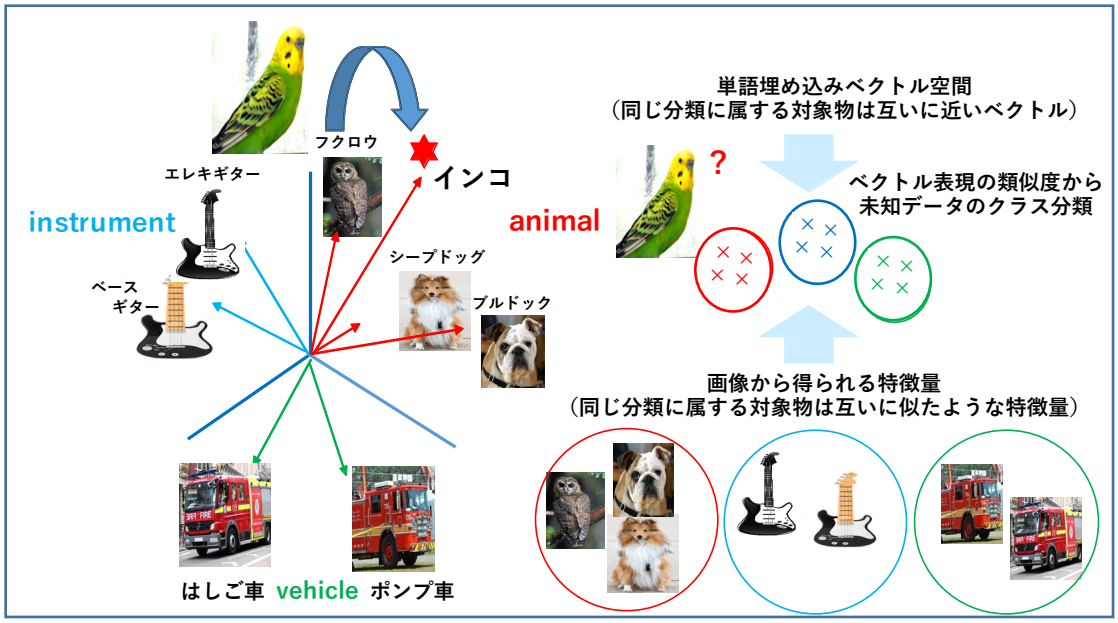


図 5 -23 ゼロショット学習の基本コンセプト

画像から得られた特徴から意味的特徴を直接マッピングする SOC[97]（Semantic Output codes），スキップグラム（Skip-Gram）を組み合わせた DeViSE[98]（Deep Visual-SEmantic model）及びオートエンコーダー（Autoencoder）注 62手法を利用して画像の特徴がラベルの分散表現と一致するように学習させた SAE[99]（Semantic Auto Encoder）といった手法がある．また，入力時（画像及び単語の埋め込みベクトルである分散表現）の特徴と出力時の特徴（ラベル）の間に互換性を持たせるために潜在変数を導入した LatEM[100]（Latent EMbeddings）及び画像の特徴と意味的特徴に共通の空間を持たせた SSE[101]（Semantic Similarity Embedding）といった手法も提案されている．

D） ワンショット学習（One-Shot Learning）

人間は，サンプルが 1 例又はごく少数例であってもその特徴の把握及び視覚的概念を学習して，新たな事例を分類する能力を有している[102]．例えば，図 5 -24 のように i）新たな事例を分類，ii）新たな事例を発想，iii）物体から関連するパーツを分類及び iv）関連した概念からの新しい概念を発想といったことが可能である[103]．1 例であっても学習による分類が可能であることから，ワンショット学習と呼ばれ，抽象的な知識の転用がワンショット学習の経路であると言われている[104]．既に学習済みの知識や特徴を用いて新しい概念を分類するという意味で転移学習の一つと考えられる．多くの教師付きデータを収集することが不可能な場合など学習データが 1 例又はごく少数の場合に利用できる手法である．

注62オートエンコーダーは，入力から特徴量を抽出し，特徴量に基づいて入力と同じものを出力するというディープニューラルネットワークのアルゴリズムの一種である．

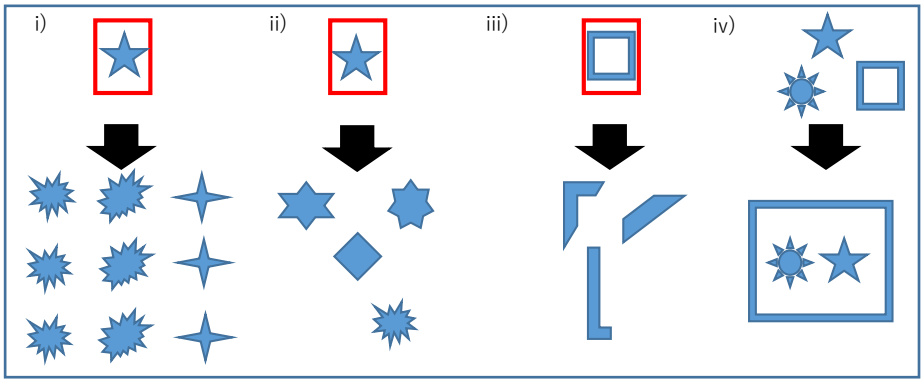


図 5 -24 ワンショット学習

入力画像はニューラルネットワークを通して特徴ベクトルとなる．2 つの同じ画像を同じニューラルネットワークにて学習すると特徴ベクトルは同一となる．この性質を利用したワンショット学習の代表的な手法としてシャムニューラルネットワーク（Siamese Neural Network）がある[105]．既知のカテゴリに属する画像とペアとなる可能性がある全ての画像について学習することで，類似性スコアを考慮して同じカテゴリに属するかどうかを予測する．このコンセプトは，創薬の分野にも応用されている．従来は，既知の分子構造から得られる特徴ベクトルと新しい化学構造を有する化合物の特徴ベクトルを比較して活性分類を行うことで，毒性や特長的な有害事象の予測が行われてきた．従来方法の代替的なアプローチとして，ワンショット学習を利用したインタラクティブ・リファインメント LSTM（Interactive Refinement Long Short-Term Memory：IterRefLSTM）といったモデルが提案されている[106,107]．

E） マルチタスク学習（Multitask leaning）

マルチタスク学習は，関連する複数のタスクで情報を共有し，タスク間の共通点や相違点を学習させることで，タスクの予測精度を向上させる帰納的転移に基づく手法である[108,109]．マルチタスク学習では，最初から複数のタスクの課題を解くためのモデルを設定する．共通の入力に対し，各タスクに対応した出力を行うことでマルチタスクが可能となる．それぞれのタスクが擬似的に教師データとして機能するため，単一のタスクによる過学習を回避でき，より汎用的な特徴表現を獲得することができる[108]．自然言語処理，画像認識及び音声認識など様々な分野で利用されている[110,111]．たとえば，自動文書要約において，重要文抽出の学習に加え，文書分類の学習を同時に行うことで重要文抽出をサポートするマルチタスク学習モデルも提案されている[112,113]．

マルチタスク学習を実行する方法として，ハードパラメータシェアリング（Hard parameter sharing）及びソフトパラメータシェアリング（Soft parameter sharing）といったアプローチがある[108]．図 5 -25 のように，ハードパラメータシェアリングは，共通した層と個別の層で構成され，下層で共通な特徴を学習する．ソフトパラメータシェアリングは，

タスクごとに個別のネットワークを持ち，各層で特徴が近くなるように学習する．ハードパラメータシェアリングは，最も一般的に用いられているアプローチである．

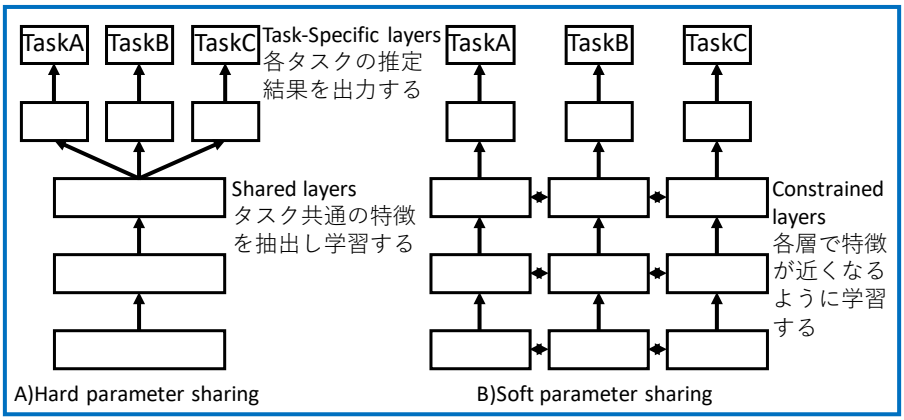


図 5 -25 マルチタスク学習のアプローチ

2017 年 10 月に，アルファ碁ゼロの仕組みを解説した論文が公表された[114]．**図 5 -26** のように，アルファ碁ゼロで用いられるデュアルネットワークにおいても，マルチタスク学習が利用されている．次の一手予測と勝率予測のモデルを一部共有することで，複数タスクの認識に必要となる共通の構造の学習が進みやすくなる[115]．

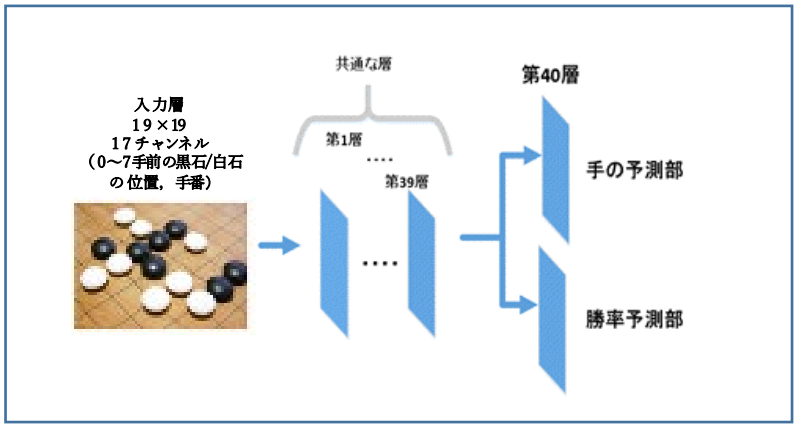


図 5 -26 アルファ碁ゼロの仕組み[115]

第 4 節 ファインチューニング（Fine-Tuning）

ファインチューニングは，目的とするタスクのデータを用いて，学習済みモデルの一部の重みを学習させ，目的とするタスクへ適合させる手法である[89]．学習済みモデルを利用したファインチューニングを行うことで，汎化性能の向上が報告されている[91]．特徴抽出（Feature Extraction）は，目的とするタスクのデータで全結合層部分を学習させる方法であったが，ファインチューニングは，畳み込み層などの中間層と全結合層部分の両方を学習させる[89]．特徴抽出のセクションでも説明したとおり，入力層から出力層へ進むに連れて

画像に対する汎用的な特徴は減り，これとは逆にデータに依存した具体的な特徴に関する情報が増える．このため，目的とするタスクのデータ出力に近い畳み込み層を全結合層と共に学習させることで，データに依存した具体的な特徴を微調整することができる．「図 5 -19 転移学習が可能なケース 3」のようなソースタスクとターゲットタスクのデータの内容が著しく異なる場合や十分なデータを集められない場合であっても学習した知識（汎用的な特徴）を活かしつつ，データに依存した具体的な特徴を微調整することで，大量のデータを用いてゼロからモデルを構築するよりも，効率的に一定の予測性能を有するモデルの実現が可能となる．

学習済みモデルのセクションでも紹介した ImageNet という日常の生活の中で遭遇する楽器や動物といった画像で事前に学習した VGG19 について，皮膚がんデータを用いてファインチューニングを行い，皮膚がんを判定させるモデルの構築を試みた．このモデル構築過程を通してファインチューニングの具体例を紹介する．なお，皮膚がんには，悪性黒色腫（メラノーマ），有棘細胞がん，基底細胞がんを始めとした多くの種類あり，その特徴や症状も様々である[116]．皮膚科専門医による良性又は悪性の鑑別精度は約 75%から 80%といった報告がある[117]．

1)　本実装で用いた画像データ

皮膚がんデータについては，Kaggle にて公開されている良性のほくろ又は悪性の皮膚がんの画像（https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign）[注 63]を利用した．学習データ，検証データ及びテストデータについて，以下のように，それぞれ benign:0, malignant:1 と教師付きラベルを施した．

- 良性のほくろ

  学習データ：200 画像，検証データ 180 画像，テストデータ：200 画像

- 悪性の皮膚がん

  学習データ：200 画像，検証データ 180 画像，テストデータ：200 画像[注 64]

更に，学習データが少ない場合は過学習を発生させる可能性が考えられることから，学習データに対して，図 5 -27 のようなデータ拡張（Data Augmentation）を行った．データ拡

注63 これらデータの全ての権利は，ISIC（International Skin Imaging Collaboration）という皮膚画像のオープンソースのパブリックアクセスアーカイブで，教育や自動診断システムの開発とテストのための画像の公開リソースに帰属している．
https://www.isic-archive.com/#!/topWithHeader/wideContentTop/main

注64 皮膚がんデータは，開発データ（良性のほくろ：1440 画像，悪性の皮膚がん：1197 画像）及びテストデータ（良性のほくろ：360 画像，悪性の皮膚がん：300 画像)が用意されている．ファインチューニングを用いて，少ないデータでも効率的にモデル構築が可能である実例を示すため，本解析用に，開発データの中から，学習用データ（良性のほくろ：200 画像，悪性の皮膚がん：200 画像）及び検証データ（良性のほくろ：180 画像，悪性の皮膚がん：180 画像）並びにテストデータの中から，性能評価には十分な量のテストデータ（良性のほくろ：200 画像，悪性の皮膚がん：200 画像）を利用した．

張は，学習データからランダムな変換を繰り返して疑似画像を作成することで，学習データを水増しする手法である．こういった手法を用いることで，データの分布について，学習データを用意することが難しい様々な状況にを考慮した学習が可能であり，汎化性を得られるようになるため過学習を回避することができる．

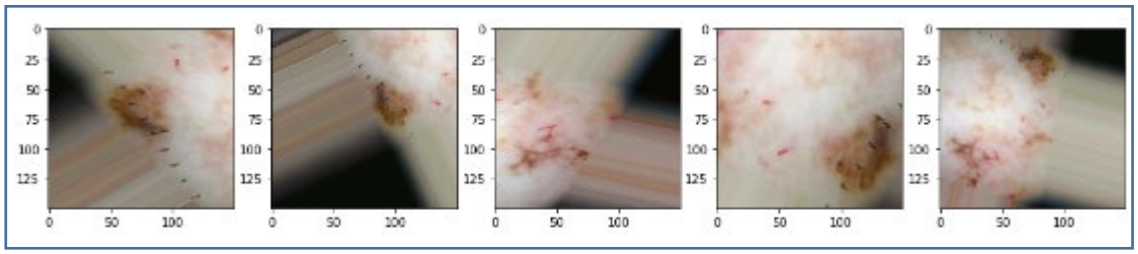


図 5 -27 学習用画像のデータ拡張

2)　ファインチューニングによるモデル構築

ソースタスク（ImageNet による画像分類）とターゲットタスク（皮膚がんの判定）のデータの内容が著しく異なることから，図 5 -28 のように，汎用的な特徴に関する重みを利用しつつ，データに依存した具体的な特徴を微調整するため VGG19 のデータ出力に近い畳み込み層（block5）を解凍[注 65]した．

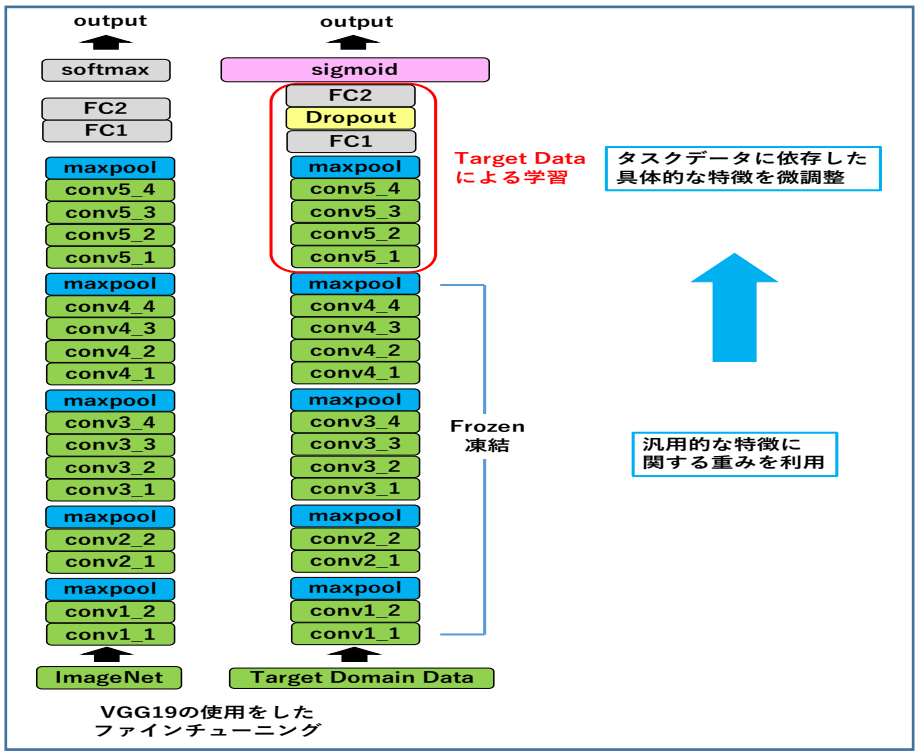


図 5 -28 ファインチューニングによるモデル構築

注65 ターゲットデータにて学習させるために該当する層を開放すること．

また，全結合層部分で，ドロップアウト（Dropout）という手法を用いた．ドロップアウトは，代表的な正則化手法で，過学習を避けるために，学習時にニューラルネットワークの一部のニューロンをランダムに一定の割合で除外（無効）する手法である．ドロップアウトの割合は，通常は 2 割から 5 割とされている．本実装では 1 つめの全結合層に対して，5 割のドロップアウトを実行し，出力ベクトルの一部を 0 と置き換え，新たなパターンを学習させることで過学習の低減を図った．出力層は，それぞれ benign:0, malignant:1 の二値データであるため，シグモイド（Sigmoid）関数とした．

3) 実装結果

学習データ 400 画像を 1 バッチあたり 80 画像，検証データ 360 画像を 1 バッチ 72 画像とし，5 ステップを 1 エポックとして，これを 20 回繰り返すミニバッチ学習を行った[注 66]．ミニバッチ学習は学習させるデータの順序による影響や外れ値によるノイズの影響を受けにくいといったメリットがある．

学習データ及び検証データの学習過程を図 5 -29 に示す．なお，学習結果は早期中止（Early Stopping）[注 67]を考慮した正解率及び損失値として二値交差エントロピー（binary cross entropy）[注 68]を用いて評価した．最終エポックにおける学習データ及び検証データの正解率は，おおよそ 87%及び 78%であった．学習データ及び検証データの誤り率は，おおよそ 29%及び 47%であり，学習データ及び検証データの結果に一部で乖離した部分があり，エポックが進むにつれて，過学習となる傾向であった．

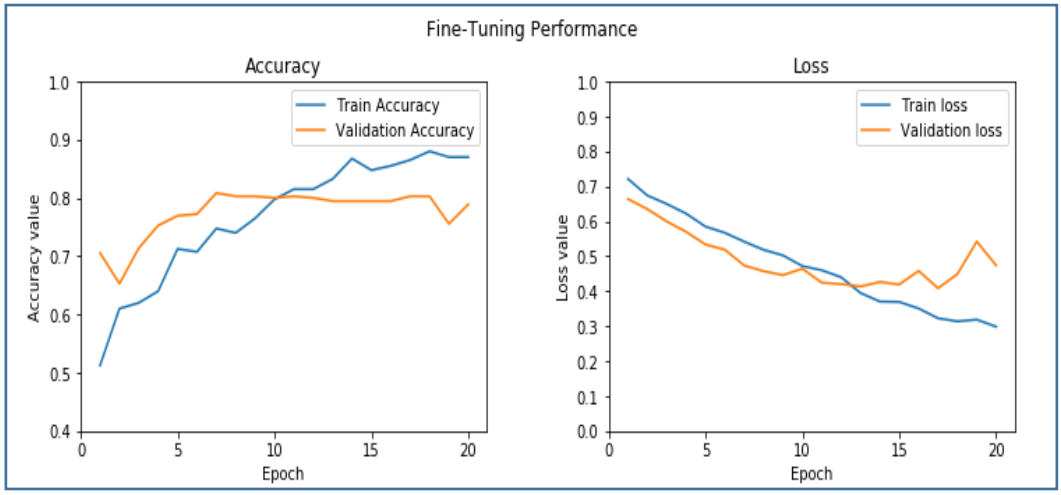


図 5 -29 学習データ及び検証データの学習過程[注 69]

注66 1 エポックあたり全データ数=バッチ*ステップ数となり，学習データ 400 画像，検証データ 360 画像をカバーすることになる．ミニバッチ学習については第 3 章第 2 節第 3 項参照

注67 学習を進めていくと学習データでは誤差が小さくなるが，検証データにおいては，誤差が大きくなる場合があり，この誤差の上がり始めを過学習し始めと考えて学習を止める手法．

注68 判定結果の正誤を判定し，その誤差を評価する関数を損失関数と呼ぶ．本実装では，出力は二値データであるため，損失関数の一つである二値交差エントロピー（binary cross entropy）をもちいた．

注69 Loss は損失関数の値．学習では，この値が小さくなると，学習ができていることを意味し，検証では汎化性能が得られていることを意味する．

学習データ，検証データ及びテストデータにおける構築したモデルによる判定結果を表5-7に示す．

表5-7 学習データ，検証データ及びテストデータにおける判定結果

学習データ		Predicted	
		benign	malignant
Actual	benign	168	32
	malignant	8	192
検証データ		Predicted	
		benign	malignant
Actual	benign	162	18
	malignant	58	122
テストデータ		Predicted	
		benign	malignant
Actual	benign	172	28
	malignant	41	159

学習データ，検証データ及びテストデータにおけるモデルの性能評価結果及びROC曲線を表5-8 及び図5-30 に示す．モデルの性能評価指標は，学習データと検証データとの間で，正解率，適合率，再現率及びF値について，おおよそ11%の乖離が認められているが，学習データとテストデータとの間では，おおよそ7%と乖離は小さくなっており，過学習の影響は限定的であると考えられた．また，テストデータにおける正解率は0.827であったことから，皮膚科専門医による良性又は悪性の鑑別精度に迫る予測精度であった．

表5-8 ファインチューニングによるモデルの性能評価結果

	正解率	適合率	再現率	F値	AUC
学習データ	0.900	0.905	0.900	0.899	0.96
検証データ	0.788	0.803	0.788	0.786	0.90
テストデータ	0.827	0.828	0.827	0.827	0.91

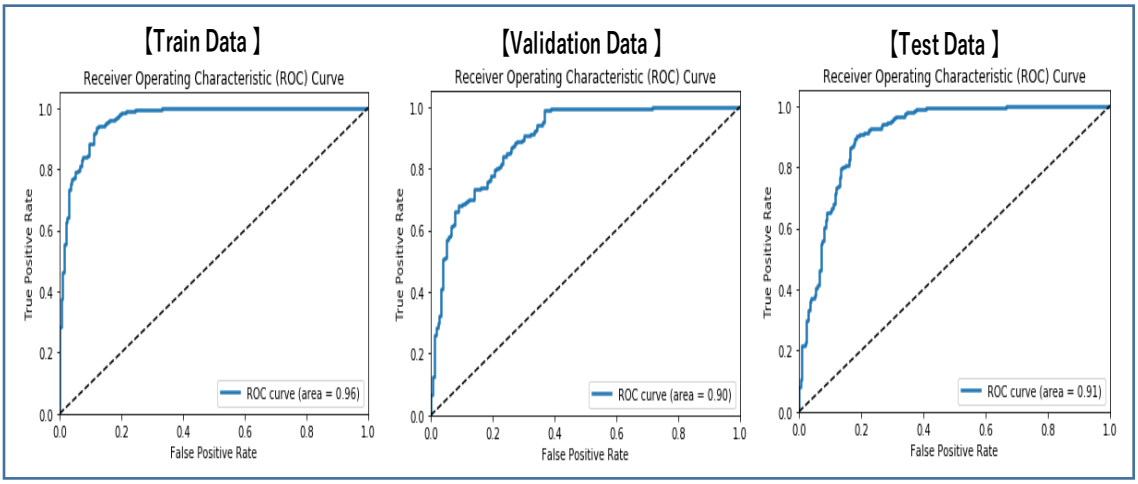


図 5 -30 ROC 曲線

大量のデータを用いてゼロからモデルを構築するのではなく，本実装のように，学習済みモデルを用いてファインチューニングを行うことで，良性のほくろ又は悪性の皮膚がん，それぞれ 200 画像の学習データでも効率的に一定の予測性能を有するモデル構築が可能となる．

第 5 節 医療分野における転移学習の活用事例

第 1 項 画像認識分野：理化学研究所の高精度緑内障自動診断装置[68]

「第 2 節 学習済みモデルの紹介」でも紹介したとおり，理化学研究所と東北大学の共同研究グループは，眼底検査装置からのマルチモダリティ画像情報を用いて，緑内障を自動診断できる機械学習モデルを構築した．

緑内障 208 眼と健常 149 眼を対象とし，視神経乳頭部のカラー眼底写真，3 次元光干渉断層計（optical coherence tomography：OCT）データから得られる視神経乳頭神経線維層の層厚マップとそのデビエーションマップ，同じく 3 次元 OCT データから得られる黄斑の神経節細胞複合体層の層厚マップとそのデビエーションマップの 5 種類の画像が緑内障を自動診断する機械学習モデルの構築に用いられた．（図 5 -31）．

デビエーションマップとは，正常眼データベースから算出した層厚マップの平均画像との差分画像のことである．これらの 5 種類の画像情報に対して，VGG19 を用いた転移学習とランダムフォレストを組み合わせることにより，少数の情報から緑内障の自動診断を行う機械学習モデルが構築された．

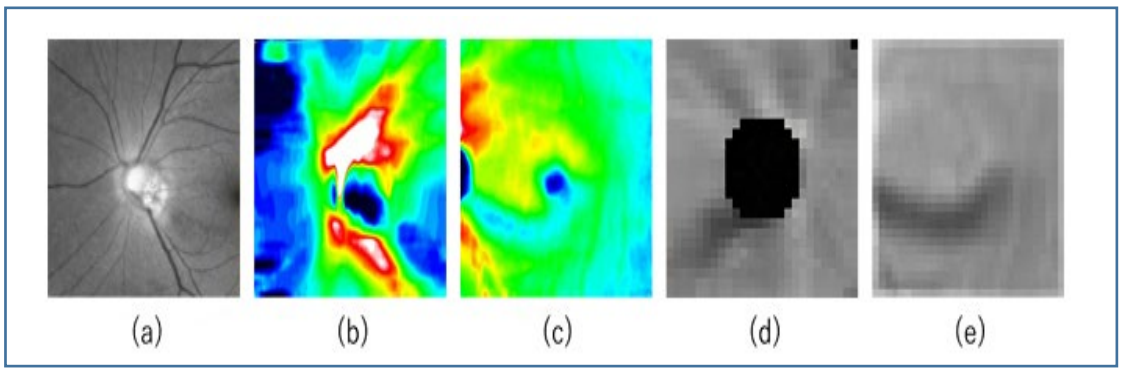


図 5 -31 機械学習に利用した 5 種類の入力画像サンプル[注 70]

(a)カラー眼底写真（b)視神経乳頭神経線維層の層厚マップ（c)視神経乳頭神経線維層のデビエーションマップ
(d)黄斑の神経節細胞複合体の層厚マップ（e)黄斑の神経節細胞複合体のデビエーションマップ

1)　転移学習の実施

畳み込みニューラルネットワーク（CNN）モデルの一種である VGG19 を利用し，正常眼又は緑内障眼を分類するために，VGG19 の出力層を 2 つのユニット（分類したいクラスの数に相当）を持つ新しいソフトマックス層に変更し，5 種類の画像データセットに対して，それぞれの VGG19 を構築した．学習過程において，ランダムな画像の左右反転，画像の回転及び水平方向へのシフトを伴うデータ拡張（data argumentation，画像を加工し数を増やす手法）並びにドロップアウトが実施された．

2)　特徴量の組み合わせの探索

図 5 -32 に示したように．各入力画像から得られた 4096 の特徴量ベクトルを用いて正常眼又は緑内障眼を分類させるために学習はランダムフォレスト（RF; tree number=10,000）により行われた．その結果，2 番目の全結合層（Fully Connected Layer）が取り除かれた．

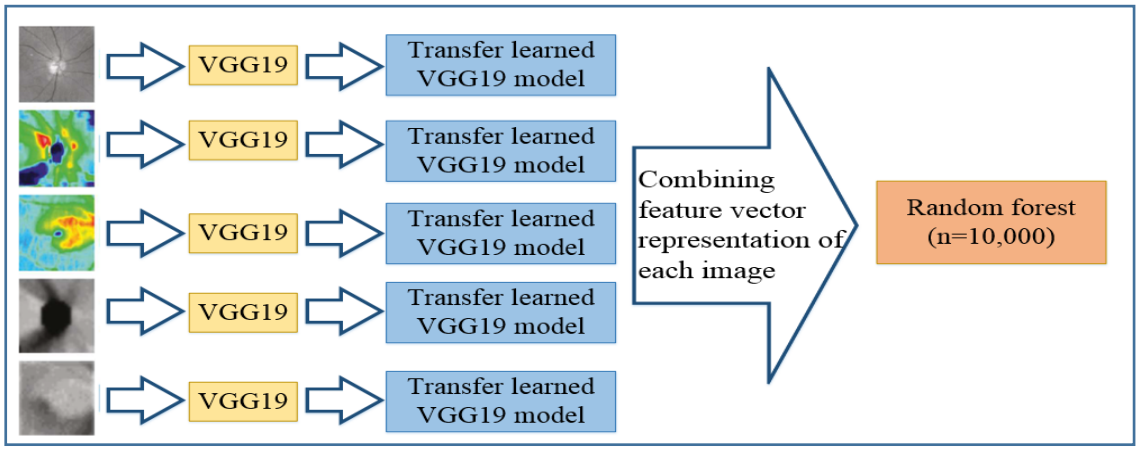


図 5 -32 緑内障の自動診断を行うための機械学習モデルの構築アプローチ[62]

注70 画像出典：http://downloads.hindawi.com/journals/jhe/2019/4061313.pdf . Open access, Creative Commons Attribution License (CC-BY).

3） 性能評価

構築した機械学習モデルの性能を 10-分割交差検証法に基づく AUC（Area Under Curve of receiver operating characteristic）を用いて評価された．単一の画像及び複数画像を統合した場合の AUC は表 5 -9 のとおりである．5 種類の画像を統合したモデル（#11）で，差はわずかであるが，最高性能として AUC=0.963 と高い診断精度が得られた．これらの結果から，構築した機械学習モデルは，早期の緑内障をより敏感に検出する可能性を持っており，緑内障の正確な診断をサポートし，日常の緑内障ケア向上に繋がると結論付けている．

表 5 -9 単一の画像及び複数画像を統合した場合の AUC

Number	Cases	AUC (mean ± SD)
#1	Disc fundus image (green channel)	0.940 ± 0.039
#2	Disc RNFL thickness map	0.942 ± 0.037
#3	Macular GCC thickness map	0.944 ± 0.032
#4	Disc deviation map	0.949 ± 0.030
#5	Macular deviation map	0.952 ± 0.029
#6	Combination of #2 and #4 (images from disc OCT data)	0.953 ± 0.032
#7	Combination of #3 and #5 (images from macular OCT data)	0.954 ± 0.031
#8	Combination of #1, #2, and #4 (images from disc OCT data with fundus image)	0.959 ± 0.031
#9	Combination of #1, #2, and #3 (automatically detected disc and macular center were not used in creating images)	0.961 ± 0.029
#10	Combination of #2, #3, #4, and #5 (images from OCT data)	0.963 ± 0.030
#11	Combination of all images	0.963 ± 0.029

第 2 項 自然言語処理分野：電子的診療記録からの疾患診断コードの自動付与[118]

医療管理の専門家や診察記録の内容をコーディングする医療事務員が，電子的診療記録（Electronic medical records, EMRs）から手作業で国際疾病分類（International Statistical Classification of Diseases, ICD）コードを付与している．EMRs からの ICD コード付与は，米国では医療費請求に利用されており，本邦においても，包括医療費支払制度で用いる DPC コーディングに用いられている．しかしながら，手作業での ICD コードの付与は，相当な時間を要すると共にミスを起こしやすくする．本項では，転移学習を活用した深層学習による EMRs からの ICD コードの自動付与に関する事例を紹介する．

ケンタッキー大学のメディカルセンターの 2011 年から 2012 年の入退院患者の 71,463 の EMRs が用いられ（UKLarge dataset），それぞれの EMR は，ICD-9 コードが付与されている．データセットのうち，2000 EMRs を検証データセット，3000 EMRs をテストデータにランダムに分割し，残りのデータセットを学習データとし，総計 1231 の ICD コードがタスクの対象となった[注 71]．なお，いくつかのまれな事象も含むため，性能への影響を考慮して，事前に PubMed による学習を行わせた重みを利用した転移学習が行われた．データセットの特徴は表 5 -10 のとおりである．

注71 ICD-9 の項目（コード）数は約 7,000 個ある．

表 5 -10 データセットの特徴[注 72]

	PubMed	UKLarge
# Instances	1,600,000	71,463
# Lables	27,150	1231
Label cardinality	12.62	7.4
Avg # words per instance	147	5303
# Code combinations	-	60,238

1)　転移学習の実施

電子的診療記録からの ICD コードの自動付与に関する方法は, 図 5 -33 に示したように, ソースタスクである PubMed による事前学習過程及びターゲットタスクである EMRs データによる転移学習を含む学習過程の 2 つの手順から構成されている.

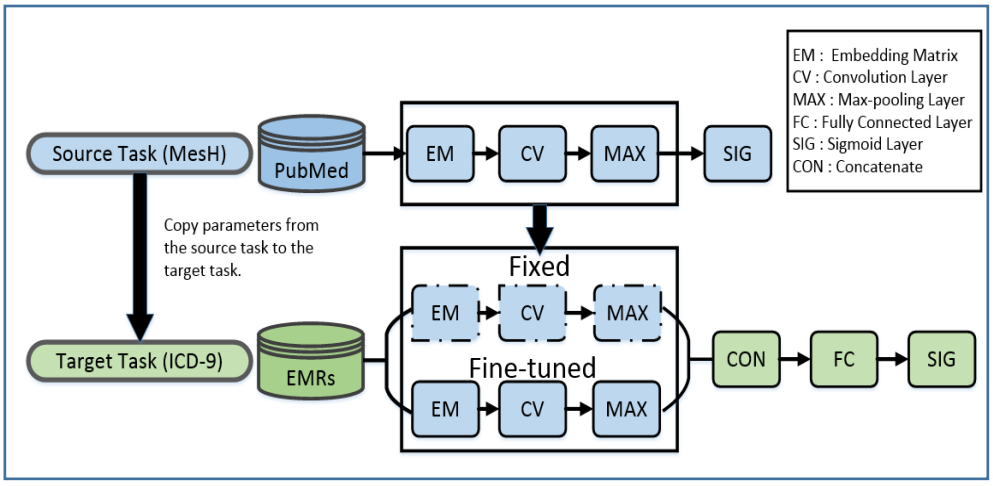


図 5 -33 電子的診療記録からの ICD コードの自動付与に関する方法

ソースタスク及びターゲットタスクのモデルとして，畳み込みニューラルネットワーク（CNN）が用いられた．ターゲットタスクでは，以下の複数の組み合わせが検討され，学習過程において，ドロップアウトが実施された．

EM[X], CV[X]：単語埋め込み及び畳み込みのいずれの重みも更新しない
EM[X], CV[レ]：単語埋め込み部分は更新せず，畳み込み部分のみファインチューニング
EM[レ], CV[レ]：単語埋め込み及び畳み込みのいずれの部分もファインチューニング

2)　性能評価

構築したモデルの性能は，Macro-F Score 及び Micro-F Score にて評価された．

注72 Instance は文献およびカルテ，#Labels は付与されている病名の単一化した数，Labeles cardinality は 1 Instances 当たりの Label の平均数，Code combinations は患者の疾病を表現するための ICD-9 コードの組み合わせ.

$$P(T_j) = \frac{TP_j}{TP_j + FP_j}, \qquad R(T_j) = \frac{TP_j}{TP_j + FN_j}, \qquad F(l_j) = \frac{2P(l_j) \cdot R(l_j)}{P(l_j) + R(l_j)}$$

$$Macro - F = \frac{1}{L}\sum_{j=1}^{L} F(l_j), \qquad P^{mic} = \frac{\sum_{j=1}^{L} TP_j}{\sum_{j=1}^{L}(TP_j + FP_j)}, \qquad R^{mic} = \frac{\sum_{j=1}^{L} TP_j}{\sum_{j=1}^{L}(TP_j + FN_j)}$$

$$and \;\; Micro - F = \frac{2P^{mic} \cdot R^{mic}}{P^{mic} + R^{mic}}$$

$P(l_j)$: Precision, $R(l_j)$: Recall, $F(l_j)$: F Score, TP_j : Positive, FP_j : False Positive, FN_j : False Negative

表 5 -11 に示したように，EM[X], CV[X]と EM[レ], CV[レ]を平均したモデルが最も良い精度であった．また，図 5 -34 のとおり，下位 10%及び上位 10%の ICD コードでは，Macro F-Score は，EM[レ], CV[レ]に比べ，それぞれ 5%及び 2%の精度の改善が認められた．また，ロジスティックモデル（LR）に比べ，ニューラルネットワークモデルの方が高い性能を有していた．

表 5 -11 転移学習のレイヤーごとの結果

	Micro *F*-score	Macro *F*-score
EM[X] CV[X]	46.5	23.6
EM[X] CV[✓]	49.8	23.8
EM[✓] CV[✓]	53.1	24.2
EM[✓] CV[✓] + EM[X] CV[X]	53.5	26.0
EM[X] CV[X] AVG	48.3	25.5
EM[X] CV[✓] AVG	51.3	25.5
EM[✓] CV[✓] AVG	54.1	25.8
EM[✓] CV[✓] + EM[X] CV[X] AVG	56.7	28.6

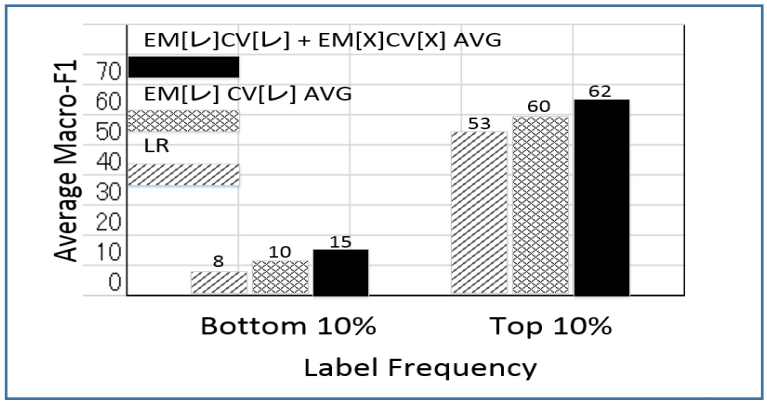


図 5 -34 下位 10%及び上位 10%の ICD コードでの性能評価

まれな事象を含む ICD コードの存在下であっても転移学習を活用することにより性能の改善が認められ，頻出 ICD コードよりもまれな ICD コードの方が性能改善の度合いが高かったと結論付けている．

表 5 -12 【代表的なフレームワークと学習済みモデル】

フレームワーク	開発	ライセンス	用途	主な学習済みモデル
TensorFlow Hub https://github.com/tensorflow/hub Google が提供している事前学習済みモデルのライブラリ．画像認識，映像認識及び自然言語処理に利用できる学習済みモデルが公開されている．	Google	Apache	画像認識	NASNet-A Inception V3 ResNet V2 50
			画像生成	Progan-128 Biggan-512 Biggan-deep-512
			映像認識	I3d-kinetcis-400 I3d-kinetcis-600
			自然言語処理	Elmo Nnml-ja-dim50 Wiki-words-250
Model Asset eXchange https://developer.ibm.com/exchanges/models/ IBM が提供しているテキスト，画像，音声，動画を処理できるオープンソース深層学習モデルライブラリ．	IBM	Apache	画像認識	Inception-ResNet-v2 ResNet-50
			音声認識	Audio Classifier Audio Embedding Generator
			自然言語処理	Text Sentiment Classifier Word Embedding Generator
Keras https://github.com/keras-team/keras TensorFlow などで実行可能な高水準のニューラルネットワークライブラリ．	Google など	MIT	画像認識	Xception VGG16 DenseNet
Caffe https://github.com/BVLC/caffe The Berkeley Vision and Learning Center が開発している深層学習用のライブラリ．	The Berkeley Vision and Learning Center	BSD	画像認識	Alexnet Caffenet Rcnn_ilsvrc

フレームワーク	開発	ライセンス	用途	主な学習済み モデル
Chainer https://github.com/chainer Deep Learning のためのフレキシブルなニューラルネットワークのフレームワーク.	Preferred Infrastructure, Inc. Preferred Networks, Inc.	MIT	画像認識	VGG16 Googlenet ResNet152
			オブジェクト検出	SSD512 YOLOv3
			セグメンテーション	SegNet PSPNet
Pytorch https://github.com/pytorch https://pytorch.org/docs/stable/torchvision/index.html 研究プロトタイピングから本番展開までのシームレスなオープンソースの深層学習プラットフォーム.	Facebook	BSD	画像認識	ShuffleNet v2 SqueezeNet MobileNet v2
			オブジェクト検出	Faster R-CNN ResNet-50 FPN Mask R-CNN ResNet-50 FPN
			セグメンテーション	FCN ResNet101 DeepLabV3 ResNet101
Cognitive Toolkit (CNTK) https://github.com/microsoft/CNTK Microsoft が提供する ニューラルネットワークを一連の計算ステップとして記述する統一された深層学習ツールキット.	Microsoft	MIT	画像認識	AlexNet GoogLeNet VGG16
MXnet https://github.com/apache/incubator-mxnet AMAZON が提供する効率性と柔軟性の両方を重視して設計された深層学習フレームワーク.	AMAZON	Apache	画像認識	DenseNet ResNet V2 SqueezeNet
PadlePadle https://github.com/PaddlePaddle/ https://github.com/PaddlePaddle/models	Baidu	Apache	画像認識	VGG ResNet Inception V4
			オブジェクト検出	SSD YOLOv3
			画像生成	DCGAN

フレームワーク	開発	ライセンス	用途	主な学習済み モデル
Baidu の科学者やエンジニアによって開発された深層学習プラットフォーム.				ConditionalGAN CycleGAN
			自然言語処理	Transformer BERT ELMo
AIRC https://www.airc.aist.go.jp/achievements/ja/ 公的な研究開発プロジェクトや民間企業との共同研究を実施し, その成果として事前学習済みモデルを公開している.	人工知能研究センター	MIT Apache	動画認識	3D-ResNets
			自然言語処理	BERT

終わりに

翻訳ソフトや会議などの音声の文字起こしなど，一般的なコンシューマ向けの，機械学習及び AI が組み込まれた，完成されたサービスの導入では，事業分野の辞書の登録などは必要であるにせよ，AI を意識することなく，通常のソフトウエアと同様に導入可能である．しかし，事業分野特有の課題解決には導入可能な完成されたサービスがあることはほとんどなく，AI を含むシステムの開発が必要となるが，他分野を含めると，機械学習としては似た仕組みが用いられている場合があり，そういった既存サービスを流用することで効率よくシステムを開発，導入することが可能となる．つまり，AI を含むシステムの開発であっても，新しい機械学習の手法を開発する必要は多くなく，既存の手法を組み合わせたり，学習済のモデルを道具あるいは部品として利用できる時代となっている．

このように機械学習自体の敷居は低くなってきた一方で，そこに用いるデータの準備が重要であることは変わらない．本報告書では第 2 章及び第 4 章でデータの準備，特徴量の作成について具体的に説明を行った．また，多くある機械学習の手法の中から適切な手法を選択する必要がある．本報告書では，基本的な方法を事例として第 4 章及び第 5 章で紹介し，R または Pyhton のコードを AppenAppendix に添付した．第 4 章ではデータ解析に機械学習を用いる場合も視野に含め，試行錯誤する部分も紹介した．第 5 章では臨床開発に応用することを想定して学習済の深層学習モデルを流用する方法を具体的事例とともに紹介した．いずれも最先端の手法や事例ではないが，その分，オープンソースのライブラリなどが充実しており，WEB 上には参考となる情報も多くあるので，興味を持った読者はとりあえず身近で簡単なこと（例えば，ロジスティック回帰分析を単層のニューラルネットワークで実行する）から始めてはいかがだろう．本書が製薬企業の特に臨床開発から市販後の場面において機械学習の手法を利用しようとするユーザーの後押しになれば幸いである．

Reference

1 日本製薬工業協会 医薬品評価委員会 「データサイエンス部会．AI ってなに?」（2019 年 5 月）
http://www.jpma.or.jp/medicine/shinyaku/tiken/allotment/what_is_ai.html

2 日本製薬工業協会 医薬品評価委員会 データサイエンス部会．「製薬企業（国内）における人工知能（AI）の利用状況及び利用への取り組みに関するアンケート 結果報告」（2018 年 12 月）
http://www.jpma.or.jp/medicine/shinyaku/tiken/allotment/use-ai.html

3 浅川伸一，江間有沙，工藤郁子，他．ディープラーニング G 検定公式テキスト．翔泳社, 2018, 329p.

4 韮原祐介．いちばんやさしい機械学習プロジェクトの教本．インプレス, 2018, 208p.

5 有賀康顕，中山心太，西林孝．仕事ではじめる機械学習．オライリー・ジャパン, 2018, 229p.

6 丸山宏．機械学習工学に向けて．日本ソフトウェア科学会第 34 回大会（2017 年度）講演論文集. 2017.

7 経済産業省. AI・データの利用に関する契約ガイドライン．経済産業省. 2018-06-15.
https://www.meti.go.jp/press/2018/06/20180615001/20180615001.html

8 Mackerel. “Mackerel における機械学習プロジェクトのマネージメント”. Mackerel. 2019-05-17. https://mackerel.io/ja/blog/entry/2019/05/17/120025, (参照 2020-02-14).

9 Simonite T. When It Comes to Gorillas, Google Photos Remains Blind. Google promised a fix after its photo-categorization software labeled black people as gorillas in 2015. More than two years later, it hasn't found one. Wired Business 01.11.2018 07:00 AM. https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/

10 知的財産戦略本部．知的財産推進計画 2017.
https://www.kantei.go.jp/jp/singi/titeki2/kettei/chizaikeikaku20170516.pdf

11 独立行政法人情報処理推進機構. AI 白書 2017．独立行政法人情報処理推進機構. 2018-3-22.
https://www.ipa.go.jp/about/report/ai/201707.html

12 Erin N. Hahn. An Overview of Open-Source Software Licenses and the Value of Open-Source Software to Public Health Initiatives. Johns Hopkins APL Technical Digest. 2014, vol. 690, no. 32, p. 690-698. https://www.jhuapl.edu/Content/techdigest/pdf/V32-N04/32-04-Hahn.pdf

13 Open Source Initiative. Open Source Initiative. https://opensource.org/

14 人工知能技術戦略会議．人工知能技術戦略（人工知能技術戦略会議　とりまとめ）．
https://www.nedo.go.jp/content/100862413.pdf

15 特許庁調整課審査基準室. AI 関連技術に関する特許審査事例について．特許庁. 2019-3.
https://www.jpo.go.jp/system/laws/rule/guideline/patent/ai_jirei.html

16 Andrew Ng. Coursera Machine Learning. https://www.coursera.org/learn/machine-learning

17 ライオンブリッジジャパン株式会社．機械学習にはどれくらいの学習データが必要か？．
https://lionbridge.ai/ja/articles/how-much-ai-training-data-do-you-need/

18 Sulman Khan. Machine Learning: Regularized Linear Regression and Bias v.s. Variance.
https://rpubs.com/SulmanKhan/445029

19 Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research 16, 321?357 (2002)

20 Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research, 61, 863-905.

21 日本製薬工業協会 医薬品評価委員会 データサイエンス部会から「臨床試験の個別被験者データの共有」が公表されている．http://www.jpma.or.jp/medicine/shinyaku/tiken/allotment/ctds.html

22 一般社団法人日本ディープラーニング協会（監修）．ディープラーニング G 検定．株式会社 翔泳社，2019，289p

23 Mikolov T, Chen K, Corrado G, Dean J, Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781v3 [cs.CL] 7 Sep 2013. https://arxiv.org/abs/1301.3781

24 Faruqui M, Dodge J, Jauhar S.K., Dyer C, Hovy E, Smith N.A., Retrofitting Word Vectors to Semantic Lexicons. arXiv:1411.4166v4 [cs.CL] 22 Mar 2015. https://arxiv.org/abs/1411.4166,

25 Willett P, Dissimilarity-based algorithms for selecting structurally diverse sets of compounds.J Comput Biol. 1999 Fall-Winter;6(3-4):447-57.

26 Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

27 scikit-learn 0.22.1 公式ドキュメント．sklearn.ensemble.GradientBoostingClassifier, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html?highlight=gradientboostingclassifier

28 scikit-learn 0.22.1 公式ドキュメント．sklearn.svm.SVC, https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

29 scikit-learn 0.22.1 公式ドキュメント．sklearn.model_selection.GridSearchCV https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html?highlight=gridsearchcv#sklearn.model_selection.GridSearchCV

30 Guokun Lai. "Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks". Cornell University. https://arxiv.org/abs/1703.07015

31 西村理明．糖尿病治療におけるデバイスの進歩．日本内科学会雑誌. 2018, vol. 107, no. 3, p. 586-592. https://www.jstage.jst.go.jp/article/naika/107/3/107_586/_article/-char/ja/

32 日本メドトロニック株式会社．日本メドトロニック アラート機能付きのリアルタイム CGM『ガーディアン TM コネクト システム』の提供を開始－予測のチカラで，より良い血糖コントロールをサポート－．プレスリリース 2018-12-03. https://www.medtronic.com/jp-ja/about/news/pressrelease/2018-12-03.html

33 アボット．アボット，糖尿病患者向けグルコースモニタリングシステム「FreeStyle リブレ」が日本で保険適用となることを発表．プレスリリース 2017-09-01. https://www.abbott.co.jp/media-center/press-releases/09-04-2017.html

34 Qingnan Sun. "Predicting Blood Glucose with an LSTM and Bi-LSTM Based Deep Neural Networks". Cornell University. https://arxiv.org/abs/1809.03817

35 Taiyu Zhu. "A Deep Learning Algorithm for Personalized Blood Glucose Prediction". KDH@IJCAI-ECAI 2018. The 3rd International Workshop on Knowledge Discovery in Healthcare Data. http://ceur-ws.org/Vol-2148/

36 Jianwei Chen. "Dilated Recurrent Neural Network for Short-Time Prediction of Glucose Concentration" KDH@IJCAI-ECAI 2018, The 3rd International Workshop on Knowledge Discovery in Healthcare Data. http://ceur-ws.org/Vol-2148/

37 Sepp Hochreiter. Long Short-Term Memory. Neural Computation. 1997, vol. 9, no. 8, p. 1735?1780. https://www.mitpressjournals.org/toc/neco/9/8

38 Felix A. Gers. Learning to Forget: Continual Prediction with LSTM. Proc. ICANN'99 Int. Conf. on Artificial Neural Networks. 1999, vol. 2, p. 850-855. https://ieeexplore.ieee.org/document/818041

39 M. Schuster. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing. 1997, Vol. 45, Issue: 11. https://ieeexplore.ieee.org/abstract/document/650093/similar#similar/

40 Nariman Noorbakhsh-Sabet MD. Artificial Intelligence Transforms the Future of Health Care. The American Journal of Medicine. 2019, 000:1-7 (article in press). https://doi.org/10.1016/j.amjmed.2019.01.017

41 Mak KK. Artificial intelligence in drug development: present status and future prospects. Drug Discovery Today. 2019, vol. 24, no. 3. https://doi.org/10.1016/j.drudis.2018.11.014

42 Dibyendu Dana. Deep Learning in Drug Discovery and Medicine; Scratching the Surface. Molecules. 2018, 23(9), 2384. https://doi.org/10.3390/molecules23092384

43 Chun Yen Lee. Machine learning on adverse drug reactions for pharmacovigilance. Drug Discovery Today. 2019, vol. 24, Issue 7, p. 1332-1343. https://doi.org/10.1016/j.drudis.2019.03.003

44 総務省. "3-5 人工知能と機械学習 総務省 ICT スキル総合習得プログラム". http://www.soumu.go.jp/ict_skill/

45 神嶌敏弘. 転移学習. 人工知能学会誌. 2010, vol. 25, no. 4, 572-580. http://id.nii.ac.jp/1004/00007564/

46 Sinno Jialin Pan. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering. 2010, vol. 22, Issue: 10. https://ieeexplore.ieee.org/document/5288526

47 Christian Szegedy. "Going Deeper with Convolutions". Cornell University. https://arxiv.org/abs/1409.4842

48 Karen Simonyan. "Very Deep Convolutional Networks for Large-Scale Image Recognition". Cornell University. https://arxiv.org/abs/1409.1556

49 Kaiming He. "Deep Residual Learning for Image Recognition". Cornell University. https://arxiv.org/abs/1512.03385

50 Tero Karras. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". Cornell University. https://arxiv.org/abs/1710.10196

51 Joseph Redmon. "You Only Look Once: Unified, Real-Time Object Detection". Cornell University. https://arxiv.org/abs/1506.02640v5

52 Kaiming He. "Mask R-CNN". Cornell University. https://arxiv.org/abs/1703.06870

53 Tomas Mikolov. "Efficient Estimation of Word Representations in Vector Space". Cornell University. https://arxiv.org/abs/1301.3781

54 Robin Brochier. "Global Vectors for Node Representations". Cornell University. https://arxiv.org/abs/1902.11004

55 Armand Joulin. "Bag of Tricks for Efficient Text Classification". Cornell University. https://arxiv.org/abs/1607.01759

56 Matteo Pagliardini. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. Association for Computational Linguistics. 2018, p. 528-540. https://aclweb.org/anthology/N18-1049

57 Matthew E. Peters. "Deep contextualized word representations". Cornell University. https://arxiv.org/abs/1802.05365

58 Jeremy Howard. "Universal Language Model Fine-tuning for Text Classification". Cornell University. https://arxiv.org/abs/1801.06146v5

59 Ashish Vaswani. "Attention Is All You Need". Cornell University. https://arxiv.org/abs/1706.03762

60 Jacob Devlin. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". Cornell University. https://arxiv.org/abs/1810.04805

61 磯健一. 音声認識における Deep Learning の活用. 日本神経回路学会誌. 2017, vol. 24 no. 1, 27-38. https://doi.org/10.3902/jnns.24.27

62 神田直之. 音声認識における深層学習に基づく音響モデル. 日本音響学会誌. 2017, vol. 73, no. 1, p. 31?38. https://doi.org/10.20697/jasj.73.1_31

63 河原達也. 音声認識技術の変遷と最先端. 日本音響学会誌. 2018, vol. 74, no. 7. https://doi.org/10.20697/jasj.74.7_381

64 Yusuf Aytar. "SoundNet: Learning Sound Representations from Unlabeled Video". Cornell University. https://arxiv.org/abs/1610.09001

65 Awni Hannun. "Deep Speech: Scaling up end-to-end speech recognition". Cornell University. https://arxiv.org/abs/1412.5567

66 Aaron van den Oord. "WaveNet: A Generative Model for Raw Audio". Cornell University. https://arxiv.org/abs/1609.03499

67 一般財団法人日本ディープラーニング協会. ディープラーニング G 検定公式テキスト. 2018, p155-165.

68 Guangzhou An. Glaucoma Diagnosis with Machine Learning Based on Optical Coherence Tomography and Color Fundus Images. Journal of Healthcare Engineering. 2019, Article ID 4061313, 9 pages. https://doi.org/10.1155/2019/4061313

69 Erkan Deniz. Transfer learning based histopathologic image classification for breast cancer detection. Health Inf Sci Syst. 2018, Sep 28; 6(1): 18. https://doi.org/10.1007/s13755-018-0057-x

70 Jia Qu. Gastric Pathology Image Classification Using Stepwise Fine-Tuning for Deep Neural Networks. Journal of Healthcare Engineering. 2018, Article ID 8961781, 13 pages. https://doi.org/10.1155/2018/8961781

71 Nils Gessert. Deep transfer learning methods for colon cancer classification in confocal laser microscopy images. Int J Comput Assist Radiol Surg. 2019, vol. 14, Issue 11, p. 1837?1845. https://doi.org/10.1007/s11548-019-02004-1

72 K. Simonyan. "Visual Geometry Group. Department of Engineering Science, University of Oxford". International Conference on Learning Representations, 2015. http://www.robots.ox.ac.uk/~vgg/research/very_deep/

73 Yufeng Zheng. "Breast cancer screening using convolutional neural network and follow-up digital mammography." Reserchgate. 14 May 2018. https://www.researchgate.net/publication/325137356_Breast_cancer_screening_using_convolutional_neural_network_and_follow-up_digital_mammography

74 単語らしい文字 n-gram の埋め込みによる単語の分散表現. 言語処理学会 第 24 回年次大会 発表論文集. 2018. https://anlp.jp/proceedings/annual_meeting/2018/

75 堅山耀太郎. Word Embedding モデル再訪. オペレーションズ・リサーチ 11 月号. 2017, vol.62, no. 11. http://www.orsj.or.jp/e-library/elcorsj62.html

76 鈴木潤. 単語埋め込みベクトルの圧縮法. NTT 技術ジャーナル. 2017, vol. 29, no. 9. http://www.ntt.co.jp/journal/1709/index.html

77 岡崎直観. 言語処理における分散表現学習のフロンティア. 人工知能. 2016, vol. 31, no. 2, pp189-201. https://jsai.ixsq.nii.ac.jp/ej/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=2107&item_no=1&page_id=13&block_id=23

78 前立腺肥大症診療ガイドライン. https://www.urol.or.jp/lib/files/other/guideline/08_prostatic_hyperplasia.pdf#search

79 西川仁. 自然言語処理概論 ―組み合わせ最適化の観点から―. オペレーションズ・リサーチ 12 月号. 2017, vol.62, no. 12. http://www.orsj.or.jp/e-library/elcorsj62.html

80 浅原正幸. Universal Dependencies 日本語コーパス. 自然言語処理. vol. 26, no. 1. https://www.jstage.jst.go.jp/article/jnlp/26/1/26_3/_pdf/-char/en

81 金山博. Universal Dependency に基づく多言語処理の共通化. 言語処理学会第 22 回年次大会. https://www.anlp.jp/proceedings/annual_meeting/2016/#C7-4

82 リクルートの AI 研究機関, 国立国語研究所との共同研究成果を用いた日本語の自然言語処理ライブラリ「GiNZA」を公開. 株式会社リクルート. 2019-04-02. https://www.recruit.co.jp/newsroom/2019/0402_18331.html

83 松田寛. 短単位品詞の用法曖昧性解決と依存関係ラベリングの同時学習. 言語処理学会第 25 回年次大会. https://www.anlp.jp/proceedings/annual_meeting/2019/#F2-3

84 日本医師会. ヘルシンキ宣言. https://www.med.or.jp/doctor/international/wma/helsinki.html

85 一般財団法人日本ディープラーニング協会. ディープラーニング G 検定公式テキスト. 2018, p166.

86 E. Soria. "L,Torrey & J.Shavlik(2009). Transfer Learning." Univ. of Wisconsin Machine Learning Research Group. http://pages.cs.wisc.edu/~shavlik/abstracts/torrey.handbook09.abstract.html/

87 神嶌敏弘. 変わりゆく機械学習と変わらない機械学習. 日本物理学会誌, 2019, vol. 74, no. 1. https://www.jps.or.jp/books/gakkaishi/2019/01/74-1.php

88 なぜディープラーニングがうまく学習できるのか. 日経 Robotics. 2017 年 2 月号. https://tech.nikkeibp.co.jp/dm/atcl/mag/15/ROB_backnumber/201702/

89 岡谷貴之. 画像認識のための深層学習の研究動向. 2016, vol. 31, no. 2, 169 - 179. http://id.nii.ac.jp/1004/00002037/

90 Matthew D Zeiler. "Visualizing and Understanding Convolutional Networks". Cornell University. https://cs.nyu.edu/~fergus/papers/

91 Mei Wang. "Deep Visual Domain Adaptation: A Survey". Cornell University. https://arxiv.org/abs/1802.03601

92 Saeid Motiian. "Unified Deep Supervised Domain Adaptation and Generalization". Cornell University. https://arxiv.org/abs/1709.10190

93 Kuniaki Saito. "Semi-supervised Domain Adaptation via Minimax Entropy". Cornell University. https://arxiv.org/abs/1904.06487

94 Yaroslav Ganin. "Unsupervised Domain Adaptation by Backpropagation". Proceedings of Machine Learning Research. http://proceedings.mlr.press/v37/

95 Teng Long. Zero-shot learning via discriminant representation extraction. Pattern Recognition letters. 2018, vol.109, 27-34. https://doi.org/10.1016/j.patrec.2017.09.030

96 Yuchen Guo. Synthesizing Samples for Zero-shot Learning. International Joint Conferences on Artificial Intelligence Organization. 2017. https://www.ijcai.org/proceedings/2017/

97 Mark Palatucci. Zero-Shot Learning with Semantic Output Codes. CS.CMU. https://www.cs.cmu.edu/afs/cs/project/theo-73/www/papers/

98 Andrea Frome. DeViSE: A Deep Visual-Semantic Embedding Model. http://www.cs.toronto.edu/~ranzato/publications/

99 Elyor Kodirov. Semantic Autoencoder for Zero-Shot Learning. CVPR2017. http://openaccess.thecvf.com/CVPR2017.py

100 Yongqin Xian. "Latent Embeddings for Zero-shot Classification". Cornell University. https://arxiv.org/abs/1603.08895

101 Ziming Zhang. "Zero-Shot Learning via Semantic Similarity Embedding". Cornell University. https://arxiv.org/abs/1509.04767

102 Brenden M. Lake. One shot learning of simple visual concepts. https://cims.nyu.edu/~brenden/papers/

103 Brenden M. Lake. Human-level concept learning through probabilistic program induction. Science. 2015, vol. 350, Issue 6266, p. 1332-1338. https://science.sciencemag.org/content/350/6266/1332.long

104 Oriol Vinyals. "Matching Networks for One Shot Learning". Cornell University. https://arxiv.org/abs/1606.04080

105 Gregory Koch. Siamese Neural Networks for One-shot Image Recognition. CS.CMU. https://www.cs.cmu.edu/~rsalakhu/papers/

106 Igor I. Baskin. Is one-shot learning a viable option in drug discovery? Expert Opinion on Drug Discovery. 2019, vol. 14, Issue 7, p. 601-603. https://doi.org/10.1080/17460441.2019.1593368

107 Han Altae-Tran. Low Data Drug Discovery with One-Shot Learning. ACS Central Science. 2017, vol. 3, no. 4, p. 283-293. https://doi.org/10.1021/acscentsci.6b00367

108 Sebastian Ruder. "An Overview of Multi-Task Learning in Deep Neural Networks". Cornell University. https://arxiv.org/abs/1706.05098

109 Rich Caruana. Multitask Learning. Machine Language. 1997, Vol. 28, Issue 1. https://dl.acm.org/citation.cfm?id=262872

110 Kazuma Hashimoto. "A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks". Cornell University. https://arxiv.org/abs/1611.01587

111 松井涼. Separation Multi-task Networks による顔器官点と顔属性の同時推定. 電子情報通信学会. http://mprg.jp/data/MPRG/F_group/F20181214_matsui.pdf

112 磯沼大. 文書分類とのマルチタスク学習による重要文抽出. The 31st Annual Conference of the Japanese Society for Artificial Intelligence, 2017. https://www.jstage.jst.go.jp/article/pjsai/JSAI2017/0/JSAI2017_1J14/_pdf/-char/en

113 西埜徹. マルチタスク学習による一貫性のある求人記事分類及び 職種フレーズと記事見出し文の生成. 言語処理学会 第 25 回年次大会 発表論文集（2019 年 3 月）. https://www.anlp.jp/proceedings/annual_meeting/2019/pdf_dir/P7-9.pdf

114 David Silver. Mastering the game of Go without human knowledge. Nature. 2017, vol. 550, p. 354-359. https://www.nature.com/articles/nature24270

115 渡部拓也. アルファ碁ゼロに使われているディープラーニングを解き明かす論文から詳細を紹介. CodeZine（コードジン）. 2018-07-24. https://codezine.jp/article/detail/10952

116 公益財団法人日本医療機能評価機構. "皮膚悪性腫瘍 Minds 版やさしい解説". 公益財団法人日本医療機能評価機構. https://minds.jcqhc.or.jp/n/pub/3/pub0054/G0000195/0004

117 Leonard H. Goldberg. Color Atlas of Dermatoscopy, 2nd. Arch Dermatol. 2003, 139(5):678. https://jamanetwork.com/journals/jamadermatology/article-abstract/479290

118 Anthony Rios. Neural transfer learning for assigning diagnosis codes to EMRs. Artificial Intelligence in Medicine. 2019, vol. 96, p. 116-122. https://doi.org/10.1016/j.artmed.2019.04.002

Appendix：R 及び Python のコード[注 73]

コードはクリップアイコン を右クリック-埋め込みファイルをディスクに保存 を行って開いてください．Python のファイルはセキュリティの関係で拡張子を「txt」にしています．必要に応じて「py」に変更ください．

対象	内容	コード
第 4 章第 3 節第 2 項	使用するデータを加工する．	
第 4 章第 3 節第 2 項	特徴量，手法の検討を行う．	
第 4 章第 4 節第 1 項 A)	gbm のパラメータチューニング	
第 4 章第 4 節第 1 項 B)	SVM のパラメータチューニング	
第 4 章第 7 節第 1 項	深層学習（LSTM）に用いるデータの作成	
第 4 章第 7 節第 2 項	LSTM による血糖値の予測	
第 5 章第 2 節第 1 項	手書き数字の畳み込みによる認識	
第 5 章第 2 節第 1 項 A)	VGG19 による画像認識	
第 5 章第 2 節第 1 項 B)	類似単語の抽出	
第 5 章第 2 節第 2 項	ヘルシンキ宣言の形態素解析	
第 5 章第 4 節	VGG19 のファインチューニングによる両性のほくろと悪性の皮膚がんの識別	

注73 R の実行環境は RStudio Version 1.2.5033，R version 3.6.2，Python の実行環境は Google colabo Python 3.6.9，Tensorflow 2.2，kelas 2.3.1，sklearn 0.22.2

執筆者

氏名	所属	担当
兼山　達也#	大日本住友製薬株式会社	第 2 章，第 3 章，第 4 章
叶内　正明	ノバルティス ファーマ株式会社	第 1 章
結城　美保	中外製薬株式会社	第 5 章
松木　啓典	キッセイ薬品工業株式会社	第 1 章，第 4 章，第 5 章
石橋　和士	大日本住友製薬株式会社	第 4 章
島田　文香（9 月迄）	ノバルティス ファーマ株式会社	第 4 章
徳田　純也	バイエル薬品株式会社	第 2 章
眞野　博貴	帝人ファーマ株式会社	第 4 章

#：リーダー

担当副部会長 山本 英晴　中外製薬株式会社