［機械学習工学］

3 機械学習応用システムのテストと検証

石川冬樹 | 国立情報学研究所
徳本　晋 | （株）富士通研究所

機械学習応用システムに対する品質保証の必要性

機械学習のライブラリやフレームワークの充実化により，機械学習を用いてシステムを構築すること自体は比較的容易となり，多くの企業が取り組むようになった．しかし実際に利用者，組織，社会にシステムを送り出す段階となると，動くシステムを構築したというだけでなく，その品質やディペンダビリティを説明する・保証することが求められる．機械学習の応用には自動運転における物体・状況の認識など高い品質・ディペンダビリティが求められるものもあり，後述する敵対的サンプルの存在などその品質に対する懸念も論じられている．

機械学習における品質というと，学習済みモデルに対してテストデータを用意し，精度などの性能指標を測定することをまず思い浮かべるであろう．しかし，製品・サービスとして送り出す際に考慮すべき品質の観点（品質特性と呼ばれる）としては，精度などはあくまで一側面にすぎない．またセキュリティなど多様な品質特性を考える以前に，何よりも「しっかり作った」と説明・保証できる必要がある．たとえば，意図された設計とプログラムコードの内容が合致しない，といった不具合がないことを何かしらのテスト・検証技術で確認することが求められる☆1．機械学習の技術ではなく，「機械学習工学」の技術としては，こういった精度測定に限らない，さまざまな観点からのテスト・検証技術も扱っていく必要がある．

当然ながら，従来のシステムに対しては，テスト・検証技術を中心に品質保証のための技術は盛んに研究されており，開発の現場においても（苦労・限界はありつつも）さまざまなプラクティスが確立されている．しかし，機械学習を用いた場合，これまでの技術やプラクティスが通用しないことがあり，企業の開発者からはさまざまな悩みの声が上がっている．機械学習工学研究会で行ったアンケートでは，40% の回答者が，テストや品質評価・保証について，「これまでの考え方がほとんど通用しなくなるので，根本的に異なる新たな考え方を用いる必要がある」としている☆2．これに対し，ガイドライン策定などの動きも出てきている（たとえばQA4AI コンソーシアム☆3）．またソフトウェア工学分野の多くのトップ国際会議においても，この 1, 2 年の間に関連する研究成果が発表されるようになってきた．

本稿においては，機械学習応用システムの特性を踏まえつつ，テストや品質保証に関する最新の研究動向を紹介し，今後の課題を論じる．

機械学習の特徴と品質保証への影響

機械学習を用いてシステムを開発する際には，訓練データから帰納的に（個々の事例に基づき）振舞いを生成し，ニューラルネットワークなど「モデル」と呼ばれるソフトウェア部品を得る．つまり，得られたソフトウェ

☆1 本稿では，個々の入力に対して対象を動作させ出力や状態変化を確認するアプローチをテストと呼んでいる，これに対し一定範囲のさまざまな入力に対し，数学的・論理的な解析などにより特定の性質が成り立つかを調べるアプローチを検証と呼んでいる．

☆2 機械学習工学研究会 Web サイトよりアンケート結果を閲覧可能 https://sites.google.com/view/sig-mlse/

☆3 http://www.qa4ai.jp/

ア部品の振舞いは，人間が直接的に決めたものではないということである．これは，従来のシステム開発において，人間のエンジニアが演繹的に（一般的規則に基づき）振舞いを書き下してきたのと大きく異なる．特に深層学習の場合，訓練データから得られるのはニューラルネットワークのパラメータ値であり，その意味を人間が理解できる形で捉えることは難しい（ブラックボックス性）．このように，機械学習を用いた場合，ソフトウェア部品の構築方法が異なるため，ブラックボックス性に限らず，得られたソフトウェア部品やそれを用いるシステム全体が，従来のシステムとは異なる特徴を持つ．本章では，テスト・品質保証の観点から，それらの特徴を論じる．

データの品質

前述のように，機械学習を用いた場合，ソフトウェア部品の構築方法が従来とは異なる．従来のソフトウェアにおいては，プログラムのソースコードが，最終的に開発者が送り出すものであり，かつ（コンパイル等を経て）利用者に直接届けられるものである．これに対し機械学習を用いた場合，開発者が直接的に定義するのは訓練データと学習を行うプログラムであり，利用者に届けられる機械学習モデルはそれらから間接的に得られることになる．また，品質評価や保証のためのテストにおいても，どのようなデータを用いるかが議論の中心となる．従来のソフトウェア工学においては，プログラムやそれにつながる設計モデルなどを議論の対象としてきた．これに対し，訓練データやテストデータに対する品質の管理・保証を行う方法論や，評価指標など道具立てが必要となる．

不完全さと性能指標

機械学習モデルは，原則として不完全である．期待された出力（正解）があったとして，常にそれを求めることはできず，性能に限界がある．その性能を事前に見積もることは困難であり，またできることとできないことの境界を明確に把握することはできない．この性質のため，機械学習モデルに対しては，「期待されたタスクをどれだけの状況で成功させることができるか」，という性能が最も基本的な品質特性となる．この品質特性は，正解率や精度（分類タスクの場合），あるいは最小二乗距離（回帰タスクの場合）などの指標で測定される．

機械学習技術の研究の多くにおいては，既存手法よりも性能が相対的に高いことを示すことが目標とされてきた．一方，製品・サービスの開発においては，利用者や，開発費用を出す顧客の満足が目標となる．従来のシステム開発においては，事前にシステムが満たすべき性質を仕様として定義，合意し，その充足をもって開発の完了としてきた．これに対し，機械学習応用システムの場合，事前に実現可能な精度等の性能を約束することはできない．また，精度等は 100% にはできないという前提下で，どれだけの性能があれば利用者にとって満足なのかを判断することも難しい．

加えて，精度等の性能指標は，テストデータセットに対して相対的に定まる値である．顧客あるいは開発者が用意したテストデータセットにおいて高い性能が出せたとしても，運用ではまったく異なるデータが多く現れる可能性があることを否定できない．

以上の点を踏まえ，機械学習モデルやそれを用いたシステムの開発においては，利用者や実際の運用環境も巻き込んでの試行錯誤や，「仮説と検証」の繰り返しにより，実験や運用を通して品質を定めていくことになる．従来のシステム開発においても，実現の難しさや，仕様に対する利用者や顧客の満足度などを事前に見積もることが難しいという問題は増えており，アジャイル開発など短期サイクルで開発とフィードバックを回していくことが多くなってきた．機械学習の場合も同様であるが，さらに「やってみなければ分からない」という不確かさが増している．

テスト不可能性

機械学習を用いて実現するタスクについては，さまざまな入力の可能性すべてに対し，正解や期待値（テス

トオラクル）を明確に定義できない，あるいは定義することが人手の作業を要するなど高コストであることが多い．たとえば，強化学習を用いたロボットの行動決定の場合，特定の状況でとるべき行動は１つに定まらないことが多い．自動運転のための画像認識の場合，正解は存在するが，画像をたくさん集めても被写体に「歩行者」などの正解ラベルを付与するには大きなコストがかかる（あるいは自動ラベリングなどの効率化をすると正確さの問題が生じる）．

するとまず，上記の不完全さの問題もあり，「パスしなかったテスト（出力の期待値と実際の出力値が合わなかった）により不具合を見つける」ということが難しくなってくる．訓練データや学習プログラムに不具合があったとしても，テストデータによってはある程度の精度が出てしまうかもしれず，その不具合に気づかない可能性もある．機械学習モデルに対して，従来の「単体テスト」という考え方，つまり，ある構成要素に誤りが混入したらすぐに検出する（よって誤りの混入個所の目処も付く），ということができないということである．

さらに，大量のテストにより信頼性に対する確信度を上げるということが難しくなる．従来も，正解値を都度定義することが高コストである場合はあったが，重要な性質をアサーションとして検証したり，Null 値参照など実行時エラーを検出したりするだけでも，大量のランダムテスト（あるいはより「賢い」テスト）などを行う意義はしばしばあった．機械学習を用いる場合，無数のテスト入力を（個々の正解を定義せずに）試してみても，実行時エラーなどはほぼなく，有効なテストにはならない．

説明可能性・解釈性

機械学習の利点は，欲しい出力を得る方法を明示的，演繹的に書き出せないような機能でも，訓練データを基に実現できることである．一方で，得られたモデルの振舞いは，人が定義したものではない．ある入力に対して出力が得られたとき，その出力が得られた理由は「訓練データと学習手法から示唆されたから」であり，人にとって納得できる形では原則存在しない．しかし，一部のアプリケーションでは，出力を基に人が最終的な意思決定（故障判断や与信判断など）をなす場合など，個々の出力が得られた理由を利用者が把握，納得したいことがある．

この性質から，機械学習モデルあるいはそれを含むシステム全体に固有の品質特性として，説明可能性や解釈性が注目されている．説明可能性･解釈性とは，システムからの出力を用いる人間が，その出力の妥当性について把握，納得することができる程度を表す．説明可能性や解釈性に向けた取り組みとしては，DARPA[☆4] における XAI（eXplainable AI）がよく知られる．

テストや品質評価の観点からは，説明可能性・解釈性は利用者による主観的側面があるため，利用者によるテストにより評価される．説明可能性・解釈性については，どう実現するかについての研究が非常に盛んに行われている．たとえば，解釈が容易なルール形式などのモデルを用いたり，判断に寄与した入力要素（特徴量）を示したりする手法がある．これらの実現手法については，本稿のスコープから外れるため詳細は割愛する．

敵対的サンプルと頑健性

機械学習モデルにおいては，入力の微量な変化（摂動）により出力が大きく変わることが知られている．そのような挙動を引き起こす入力を「敵対的サンプル」，そのような挙動を引き起こす入力変化を「敵対的摂動」などという．特に画像認識において，性能が非常に高い識別器においても，少しのノイズで出力を大きく変えることができることが知られている．機械学習や人工知能の信頼性や安全性，セキュリティを問う論文や記事においては，交通標識の誤認識を起こす例がよく引用される[☆5]．さまざまな摂動に対して，モデルの出力が影響されないという頑健性は，１つの重要な品質特性であると考えられている．

☆4 アメリカ国防高等研究計画局（Defense Advanced Research Projects Agency）

☆5 https://spectrum.ieee.org/cars-that-think/transportation/sensors/slight-street-sign-modifications-can-fool-machine-learning-algorithms

出力の適切さ

機械学習モデルからの出力は，人間のエンジニアがそれを得る規則を直接定義したものではなく，不適切な出力が出てこないかという懸念がある．たとえば，出力が特定の範囲に収まるという保証は原則難しく，出力の事後処理により担保する必要がある．出力の適切さには，ルールとして明文化することが難しいような多様なものがある．たとえば，給与額判断において，性別によって給与が変わってはならない，といった社会的な公平性が求められる場合もある．ほかにも翻訳によって放送禁止用語など不適切な単語が現れない，といった適切さも考えられる．

機械学習応用システムのテスト・検証の研究動向

本章では，この数年にソフトウェア工学コミュニティにて発表された研究を中心にして，テスト・検証，および品質保証のためにどのような考え方がなされているかを論じる．

カバレッジ

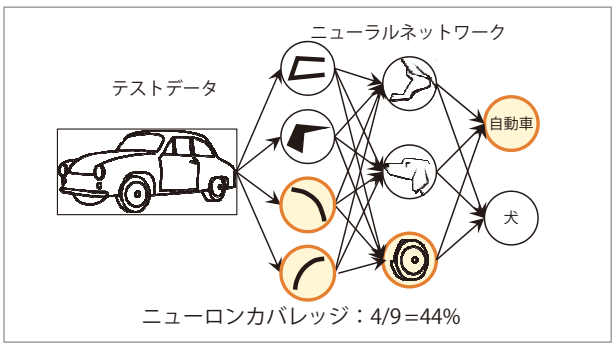
従来のソフトウェアにおいては，テストが十分になされたかの判断指標としてカバレッジ指標が用いられてきた．深層学習で得たモデル（ニューラルネットワーク）も一種のプログラムコードであるが，コードの分岐カバレッジを 100% にする（一式のテストによってすべての分岐が少なくとも 1 回ずつは実行されるようにする）ことなどは容易であり，それをもって十分にさまざまな可能性をテストしたとは言いがたい．このため，ニューラルネットワーク専用のさまざまなカバレッジ指標が提案されている（ニューロンカバレッジ [1] 等）．

ニューロンカバレッジは訓練済みのニューラルネットワークに対し，テストデータを実行することで活性化されたニューロン数の割合である．具体例を図 -1 で示す．ここではすべてで 9 個のニューロンがあり，そのうち 4 個がテストデータで活性化されたとすると，ニューロンカバレッジとしては 4/9 ＝約 44%となる．それぞれのニューロンの活性化は入力の特徴を表現しているため，ニューロンカバレッジが高いテストデータはより多様な入力の特徴を捉えていると考えられる．さらにニューロンカバレッジはさまざまなバリエーションが提案されており，ニューロンの活性化のパターンに対するカバレッジや，Modified Condition/Decision Coverage（MC/DC）と呼ばれる航空産業などで多く使われているカバレッジと同じ発想を適用したものなどがある．

テストデータの評価としては，ミューテーション解析と呼ばれる方法が使われることがある．一般的にミューテーション解析は，対象プログラムへの人為的なバグの埋め込みと，テストデータによりそのバグを検出できるかを確認することを繰り返し，検出できたバグの割合によりテストデータのバグ検出能力を評価する手法である．埋め込むバグの種類については，たとえば '+' の演算を '－' に置き換えるようなプログラム要素に対するものが従来より使われてきたが，ニューラルネットワークに対しては専用のものが提案されている．たとえばニューロンの活性化を反転させたり，レイヤの追加・削除を行う [2]．このようなモデルの異常に対し，元とは異なる分類結果となるデータが多いかを確認することで，決定境界付近に対するテストが多くされているかを評価できる．

テストデータ生成

カバレッジを上げることが機械学習応用システムの品質向上に寄与することは上で書いた通りだが，そのカバレッジを上げるには適切なテストデータを用意する必要がある．一方で画像を扱うような機械学習応用システム



■図 -1　ニューロンカバレッジの例

では生成するデータの自然さが特に重要となるため，すでにあるテストデータからの距離ができるだけ近くなるテストデータを生成することが好ましい．よって，できるだけカバレッジを上げつつ，できるだけ既存のテストデータに似たデータを用意できることが好ましい．このような問題は最適化問題として扱うことができ，従来型のシステムに対するテストデータ生成と似たようなアプローチによって，その最適化問題を解き自動的にテストデータを得ることができる（サーチベースドテスティングと呼ばれる）．探索的な方法では，元となるテストデータに少しずつ加工を加え，初めてテストが失敗したものを新たなテストデータとして生成する．このようなテストデータにはニューロンカバレッジを増加させる性質を持っている[1]．解析的な方法では，各ニューロンの振舞いをモデル化したものと，ニューロンカバレッジなどのカバレッジ基準を制約条件とし，元のテストデータからの距離が最小となるデータを最適化ソルバーで求める．生成したテストデータについては，ラベルを付けることにより訓練データとしても用いることができ，よりロバストなモデルを構築できる．代表的な実装としては DeepXplore[☆6]，DeepConcolic[☆7]，TensorFuzz[☆8] などがある．

また，カバレッジ向上以外の目的として

- 公平性に問題のあるデータを探索的に生成する
- 1ピクセルを変えるだけで分類が変わるデータを記号実行を用いて生成する

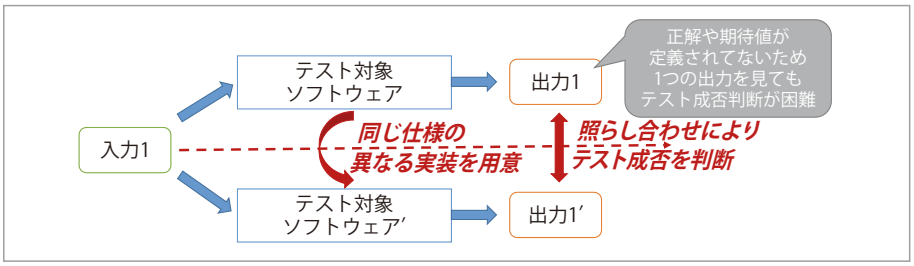
といった手法が提案されている．

疑似オラクル・メタモルフィックテスティング

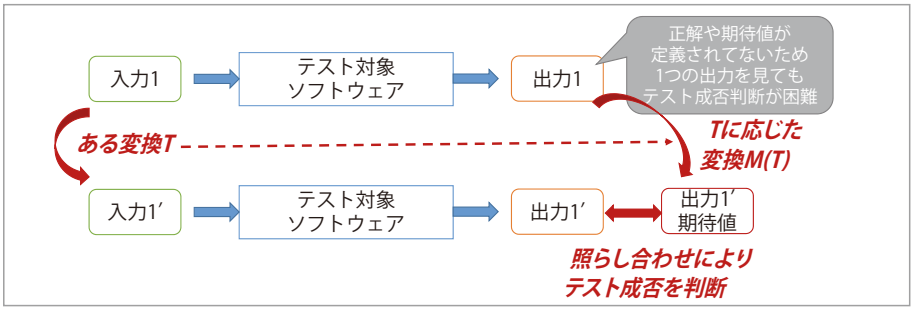
機械学習で得たモデルやそれを含むシステムに対しては，さまざまな入力の可能性すべてに対し，正解や期待値（テストオラクル）を明確に定義できない，あるいは定義することが人手の作業を要するなど高コストであることが多い．このため，テストケースの数が限られがちである．多数・多様な入力を試すテストケースを設けるために，比較対象となる疑似オラクルを用意する．たとえば，古いバージョンとの比較や，同じ機能仕様を持つ別の実装を用意し比較すること（N バージョンプログラミング（図 -2））を行う[3]．必ずしも出力が完全に合致するわけではない場合でも，誤差・距離を定義し，それが大きいケースを（自動）探索し調べることで誤りに気づいたり知見が得られたりすることがある．

メタモルフィックテスティング（図 -3）においては，「入力に対してある一定の変化を与えたときに，出力の変化が理論上予想できる」という関係（メタモルフィック

☆6 https://github.com/peikexin9/deepxplore
☆7 https://github.com/TrustAI/DeepConcolic
☆8 https://github.com/brain-research/tensorfuzz



■図 -2　N バージョンプログラミング



■図 -3　メタモルフィックテスティング

関係）をテストの成否判断に用いる[4]．たとえば，ある購買データを入力として商品の推薦ランキングを出力したとき，その出力の正しさ・妥当性を論じることは難しい．ここで元の購買データから1位の商品を削除した購買データを入力として推薦ランキングを出力すると，元は2位，3位だった商品が1位，2位に繰り上がっていることが期待される．この関係の確認であれば，任意の妥当な購買データを入力として検査することができる．モデルやシステムの特性を踏まえ，入力値へのある種の追加や除外，並び替え，逆転，一定のずらしなどを考える．表-1に機械学習応用システムに対するメタモルフィックテスティングの例をいくつか示す．注意したいのはテスト対象が訓練済みモデルだけではなく，訓練アルゴリズムであるものも含んでいる点である．ある変換 T を行った訓練データで訓練し，その訓練済みモデルにおいて同じ変換 T を行ったテストデータに対する出力の確認を行うことで，訓練アルゴリズムに対しても検査が行える．

頑健性検査

敵対的サンプルの存在を踏まえ，入力に対して一定の摂動（画像であればゆがみや照度変化，一部の欠損など）を加えても，モデルの出力が変わらないこと（頑健性）を検査する．これは単純には，ランダムに摂動を加えて画像認識の結果が変わらないことを確認するランダムテスティングとして実装できる．加えて，前述のように，認識結果が変わるような摂動の方向性を探索するサーチベースドテスティングとしてより効率的に実装することもなされている．

頑健性検査については，形式検証による網羅的検査も実現されている．すなわち，特定の画像に，一定範囲内の摂動を加えても認識結果が変わることがない，といった性質を保証する，あるいは成立しないのであれば反例を見つける．たとえば，SMT（Satisfiability Modulo Theories）ソルバーを用いて制約が満たされることを確認したり[5]，抽象解釈を用いて計算結果の取り得る範囲を効率的に近似する手法などが提案されている．

システム全体のテスト・品質

機械学習モデルの品質自体は，特に精度などが100%になり得ないことから，突き詰めてもキリがないところがある．そこで，システム全体の検証や評価を行うことが重要となってくる．たとえば，画像認識を用いて運転制御を行うシステムにおいて，「認識が失敗することで危険な状況に至る可能性」を追求する手法が提案されている[6]．逆に言うと，たとえば遠くにある先行車の認識に失敗しても衝突などが起きることはない．認識失敗の可能性をむやみに追求するのではなく，その結果危険な状況に至る可能性があるときに絞って認識失敗の可能性を追求する．そのために，システム全体の検証と誤認識を起こす画像の探索とを行き来する．

■表-1　メタモルフィックテスティングの例

アルゴリズム・アプリケーション	テスト対象	メタモルフィック関係：入力をあるやり方で変えたとき（T），出力はどうなるべきか（$M(T)$）
深層学習による自動運転	訓練済みモデル	入力画像に雨や霧に相当するノイズを追加したときのハンドル角の予測値と元の予測値との差の二乗が訓練データにおける平均二乗誤差以下であること
SVM※，深層学習による画像分類	訓練アルゴリズム／訓練済みモデル	RGBチャネルの入れ替えたデータや転置した畳み込み演算（行列）で訓練したモデルでも分類結果が変わらないこと／正規化した画像や特徴量を定数倍した画像でも分類結果が変わらないこと
クラスタリング	クラスタリングアルゴリズム	データの中心点への移動，中心点付近へのデータ追加などでクラスタリング結果が変わらないこと
時系列データ分析	訓練済みモデル	幾何学変換した出力と，幾何学変換した入力に対する出力が等しいこと
k近傍法，純ベイズ分類器	訓練アルゴリズム	アフィン変換されたデータや，複製が加わったデータセットなどで訓練したモデルでも分類結果が変わらないこと

※ Support Vector Machine

なお，この手法では，CG によるテスト画像の合成を用いているため，特定の画像と，運転制御のシミュレーションに影響する先行車との距離などの情報を紐づけることができる．

また機械学習モデルを，ユーザの体験向上やユーザによる行動促進（購入や広告閲覧など）につなげるために用いる場合，それらの KPI（Key Performance Indicator）を用いて機械学習導入前後などの比較実験を行う（A/B テスティングと呼ばれる）．

品質管理のプロセス

プロセスや管理の観点からの原則・アプローチは，研究事例ではなく，どちらかというと産業界からの経験として多く語られている．ここでもそのような原則を紹介しておく．

前もって担保できる品質（特に精度等の性能指標）を予測し合意することは難しく，実験を繰り返しながらどこまで品質を実現できるのかを探る試験的・探索的なプロセスを採ることになる．場合によっては，たとえば要求やユースケースを調節して，「適合率が低くても再現率が高ければ受け入れる」といった判断が必要となる．試験的・探索的なプロセスについて，顧客をはじめとしたステークホルダの理解・協力が必要となる．

訓練に用いたデータと運用時に入力されるデータの分布が異なる場合に，品質（特に精度等の性能指標）が劣化する場合がある．これは開発時の想定が不十分であった場合に生じるが，自動運転等実世界を扱う場合想定を尽くすことは困難である．さらに，顧客の行動や嗜好，実世界に存在する物体やそれを捉えるカメラ特性等は変化するため，もし開発時の想定が十分なされていたとしても，実行時の性能劣化があり得る．このため，実行時の監視機構を設け，性能劣化を検出するなどの対応が必要である．問題追及をより容易とするためには，結果としての性能だけでなく，入力値の範囲や分布など個々の要素やそれらの関係についての監視を行うようにする．

オンライン学習を行う，つまり実行時のデータを訓練データとして継続的にモデルを更新し続ける場合，最新の分布に適応することができる．しかしこの場合，不適切な訓練データにより不適切なモデルが得られる可能性がある．継続的な学習（モデルの更新）を自動化する際に，訓練データの選別，性能評価やテスト等も併せて自動化する必要がある．

今後の課題

ソフトウェア工学の学術界を中心に機械学習応用システムに対するテスト・検証の手法は近年多く提案されている一方，この分野は従来型の品質保証技術に比べるとまだまだ歴史も浅く，多くの挑戦すべき課題が残されている．

さまざまな品質特性

本稿では，機械学習固有の品質特性として，説明可能性について述べた．また本特集の他の記事では，機械学習応用システムにおけるセキュリティやユーザビリティに関する話題がある．このような品質特性（あるいは非機能要求）に関する包括的な議論はまだ十分になされていない．機械学習を用いた場合，従来と異なる品質特性を考える必要があるのか，また従来からある品質特性だとしてもどのように実現したり評価したりするのか，現状包括的に整理はなされていない．また訓練時のスループットや推論時の応答時間といった効率性に関する性能のテストについても同様に今後さらなる議論が必要となる．

デバッグ

ほぼすべての機械学習応用システムにおいて精度を 100% にすることはできない．つまり入力に対して常に期待する結果が出力されるわけではないことになる．従来のシステムにおいては期待する結果が出力されない場合はバグとして扱われるが，機械学習応用システムにおいてはそのようなものもある程度許容される必要がある．一方で自動運転のようなミッションクリティカル

な領域では許容されない出力結果もあり得る．まず修正すべきバグをどのように定義するべきかを考えなければならない．現状ではテストデータ生成や疑似オラクルのようなバグ検出を補助する手段はあるものの，それをバグとするかは最終的には人間の判断となる．仮に問題となり得る現象を検出したときに，次にその原因を特定しなければならない．先で述べたように機械学習応用システムでは得られた出力に対する理由を説明するのが困難であるため，問題の原因特定が非常に難しいものとなる．さらに原因特定できたとして，そのバグを取り除くことは容易ではない．なぜならすでにある訓練済みモデルに対してある入力データの影響を取り除きたい場合は訓練しなおす必要がある．また，新たな訓練データを追加して再訓練することで修正を試みる場合，追加した訓練データにより新たな問題が引き起こされる可能性があり，その予見や原因特定は上記と同様の理由で困難である．現在多くの関心が寄せられている説明可能性・解釈性についての研究は原因特定やバグ修正の問題を解決する糸口になり得ると考える．

対象機械学習手法の拡充

近年研究されている機械学習応用システムに対するテスト・検証の技術は，そのほとんどが CNN (Convolutional Neural Network) による画像分類を対象としている．そのため，RNN (Recurrent Neural Network) や LSTM (Long Short-Term Memory) などの時系列データに対するニューラルネットワークに対しての評価はほぼ行っておらず，さらに教師なし学習，強化学習，転移学習といった他の多くの学習手法に対してまで考慮されているものは数少ない．もちろんこのような手法においても本稿でこれまで説明したような機械学習応用システムの不完全さやテスト不可能性などの問題は同様に存在するため，今後取り組むべき研究課題となり得る．また，テストデータ生成技術においては，決定境界の近傍にある，人間にとっては判断しにくいような敵対的な画像データを生成するものが現状では特に多く，その他の特徴を持ったテストデータに関しては今後増えていくものと思われる．

複雑なシステム構成における保証

現存する機械学習応用システムのテスト・検証技術のほとんどは，単一の機械学習モデルを扱うシステムに対してのみ評価が行われており，より複雑なシステムに対しては考慮されていない．たとえば，複数の機械学習モデルを扱うアンサンブル学習や，振舞いが決定的であるルールベースとの組合せのシステムなどは，現実的にはシステムとして多く採用されているにもかかわらず，その品質保証については後手に回っている．

ツール・プラクティス・経験的な知見

従来のシステムに対するテストについて，その不完全さから，経験的な知見を活用することが研究においてもプラクティスとしても行われてきた．たとえば，「これ以上テストをしても，コストに見合うほど不具合は検出されなさそうだ」という判断を，不具合検出数の変化に関する経験的な知見を基に行う．あるいは，All-Pair 法など効率と網羅性のバランスをとる手法でもある程度十分に不具合が見つかる，という経験的知見も有用である．機械学習に関しても，「どれだけテストをすべきか」「あるテスト手法がどれだけ有効か」といった経験的知見が必要となるが，今はまだそのような知見がほとんど得られていない．

たとえばニューロンカバレッジは多くの論文にて使われている指標だが，その十分性についてはまだ多くの議論がされており評価は定まっていない．一方，ニューロンカバレッジを用いたものをはじめ，本稿で紹介した手法の実装は GitHub などで公開されていることが多いが，研究用途としては特定のデータセットのみを対象とするため，汎用的に使えるツールとして完成されているものは少ない．そのような背景もあり，産業界においての適用事例がまだほとんど存在しておらず，多くの手法について産業用のデータを用いての妥当性検証はされていない．

現在のソフトウェア工学分野においてはより広く，

Empirical Software Engineering と呼ばれる，現実の観測・分析に重きを置く領域での研究が盛んになっている．機械学習については，演繹的・論理的なアプローチが難しい点もあるため，汎用性が高いツールの開発とその利用を通し，さまざまな経験的知見が蓄積，分析され，利用可能になることが望ましい．

テストの計画・設計・体制に関する指針

テストの計画・設計・体制に関する指針や，その指針を支えるような実証データは現状あまりまとまっていない．本稿で述べたように不確定要素が多く試行錯誤が必要だからといって，何も事前に決めないというわけにもいかないだろう．人員配置やコストの計画もあるであろうし，検査基準を決めず結果を見てしまうと迎合してしまうといったこともあるかもしれない．

そもそもの原則も確立していない．たとえば，テストデータは訓練データとは別に存在すべきか，という問いがある．せっかくデータがあるなら訓練にも使いたいという意見も聞く．一方，開発・構築を担当するエンジニアとは別に，検査あるいはリリース受け入れ判断をするエンジニアがいて，独立した立場で分析してテストデータを用意するという考えもある．独立した 2 つの視点から取り組み突き合わせることで，特定の思い込みを排除するなど，問題点に気づくことができる可能性があるためである．ここで，そのようなテストデータがあった場合，もしも何度もテストを実行できるのであれば，訓練によってそのテストを通るようにモデルを（過度に特化して）チューニングしてしまうことがいくらでもできるかもしれない．すると，リリース受け入れ判断に使うようなテストデータは，1 回だけの受入れ妥当性確認に使うと考える立場もあるかもしれない．こういった考え方が，現状は各エンジニアの「想い」として存在しており，その共有や，それぞれの考え方の有効性に関する実証的評価が今後必要となるであろう．

今後の展望

本稿で述べたように，機械学習応用システムのためのテスト・検証技術の研究はこの 1, 2 年で盛んに発表されるようになってきたが，製品・サービスを安定して送り出すための品質保証のための課題はまだまだ多い．機械学習（アルゴリズム）自身の実践と同様に，新しい技術に果敢に踏み込みつつ，産業界における実データや実践的評価観点を含めての先端的な研究開発が必要不可欠である．個々の企業・研究者の奮闘だけに頼らず，機械学習工学研究会という場や産学の対話などを通した連携により切り拓いていければと思う．

参考文献

1) Pei, K., Cao, Y., Yang, J. and Jana, S. : Deepxplore: Automated Whitebox Testing of Deep Learning Systems, In Proceedings of the 26th Symposium on Operating Systems Principles, ACM, pp.1-18 (Oct. 2017).
2) Ma, L., Zhang, F., Sun, J., Xue, M., Li, B., Juefei-Xu, F., ... and Wang, Y. : DeepMutation : Mutation Testing of Deep Learning Systems, arXiv preprint arXiv:1805.05206 (2018).
3) Srisakaokul, S., Wu, Z., Astorga, A., Alebiosu, O. and Xie, T. : Multiple-Implementation Testing of Supervised Learning Software, In Proc. AAAI-18 Workshop on Engineering Dependable and Secure Machine Learning Systems (EDSMLS) (2018).
4) Chen, T. Y., Kuo, F. C., Liu, H., Poon, P. L., Towey, D., Tse, T. H. and Zhou, Z. Q. : Metamorphic Testing : A Review of Challenges and Opportunities, ACM Computing Surveys (CSUR), 51 (1), 4 (2018) .
5) Huang, X., Kwiatkowska, M., Wang, S. and Wu, M. : Safety Verification of Deep Neural Networks, In International Conference on Computer Aided Verification, Springer, Cham, pp.3-29 (July 2017).
6) Dreossi, T., Donzé, A. and Seshia, S. A. : Compositional Falsification of Cyber-physical Systems with Machine Learning Components, In NASA Formal Methods Symposium, Springer, Cham, pp.357-372 (May 2017) .

（2018 年 9 月 3 日受付）

■**石川冬樹（正会員）** f-ishikawa@nii.ac.jp

国立情報学研究所 アーキテクチャ科学研究系 准教授．電気通信大学 情報理工学研究科 客員准教授．博士（東京大学，情報理工学，2007 年）．形式手法や最適化技術の応用を中心に，ソフトウェア工学および自律・スマートシステムに関する研究に従事．

■**徳本　晋（正会員）** tokumoto.susumu@jp.fujitsu.com

（株）富士通研究所 ソフトウェア研究所 研究員．早稲田大学 招聘研究員．2004 年東京大学大学院情報理工学系研究科修士課程修了．2017 年同大学院博士課程単位取得退学．主にソフトウェア工学，ソフトウェアテストに関する研究開発に従事．