なぜ AI ソフトウェアのテストは困難なのか

Nippotica Corporation

AI システムは、データから学び、明確な仕様がなく、予測不可能な結果を生み出すため、伝統的なソフトウェアよりもテストが難しい。本論文では、なぜ伝統的なテスト方法が不適切であるかを説明し、不確定性、データバイアス、隠れたユーザーの期待といった主な挑戦を取り上げます。また、メタモルフィックテストや対抗的テストといった新興技術を紹介し、AI ソフトウェアを効果的にテストするための実践的な 5 つの原則で締めくくられます。

AI systems are harder to test than traditional software because they are designed to learn from data, lack clear specifications, and often produce unpredictable results. This paper explains why conventional testing methods fall short and highlights key challenges such as non-determinism, data bias, and hidden user expectations. It also introduces emerging techniques like metamorphic and adversarial testing, and concludes with five practical principles for testing AI software effectively.

対象読者：

AI 開発者、品質保証（QA）エンジニアリングチーム、ならびに AI システム展開を担当するプロダクトマネジメント専門家。

1　序章：AI の進化とテストの謎

人工知能（AI）は、今や私たちの日常生活の一部となっています。スパムメールをフィルターしたり、商品をおすすめしたり、バーチャルアシスタントを動かしたり、医師が病気を診断するのを助けたりもします。しかし、この強力な技術の裏には、ソフトウェアエンジニアが日々直面している問題があります。それは、AI ソフトウェアが信頼性があり、公平で、安全に動作するように、どのようにテストすれば良いのかという課題です [1]。

ソフトウェアのテストは、目新しいものではありません。何十年にもわたり、エンジニアは従来型のプログラムを、入力 A に対して出力 B が得られるかどうかを確認することでテストしてきました。しかし、AI はそれとは異なります。AI はデータから学習し、予測が難しく、ときには動作しているように見えても、誰にも説明できない理由で誤った結果を出すことがあります。本ブログでは、なぜ AI ソフトウェアのテストがそれほど難しいのか、そしてなぜ従来のテスト手法が十分ではないのかについて解説します。

2　AI ソフトウェアの開発プロセス

従来のソフトウェアとは異なり、AI システムでは、モデルはトレーニングデータから学習することで作成され、これがモデルの構造と内部パラメータの両方を形作ります。

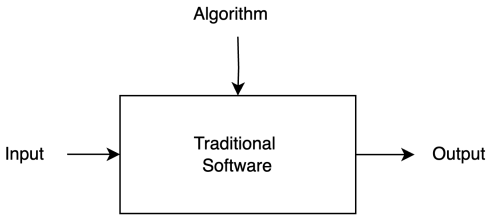

図 1　従来のソフトウェア構造

その結果、AI ソフトウェアは通常、2 つの異なるフェーズで動作します：

トレーニングフェーズ： このフェーズでは、既知

の入力とその期待される出力を使用して、いくつかの候補から最も効果的なモデルを構築、テスト、選択します。

推論フェーズ： トレーニングが完了すると、選択されたモデルがデプロイされ、新しい入力に基づいて予測や意思決定を行います。ただし、これらの出力が常に正確とは限りません。時間の経過とともにパフォーマンスを向上させるため、多くのアプリケーションでは、結果が正しくないと思われる場合にユーザーフィードバックを収集する仕組みが組み込まれています。

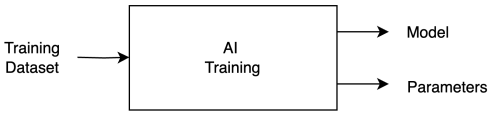

図 2　AI トレーニングフェーズ

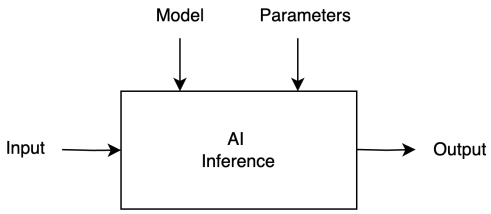

図 3　推論フェーズ

3　AI ソフトウェアは予測不能

従来のソフトウェアとは異なり、AI システムは同じ入力に対して異なる結果を出すことがあります。これは、学習中のランダム性や、モデルの最適化方法の違いによって生じます。たとえば、画像認識システムが、同じ写真を学習方法によって異なるラベルで分類することがあります。この予測不能性により、明確な期待結果を持つテストケースの作成が難しくなります。入力が変わっていないのに、ある日は合格して、別の日には不合格になることもあります [2]。

このような非決定性は、ソフトウェアテストの世界では「オラクル問題」と呼ばれます。従来のソフトウェアでは、正しい出力（オラクル）が事前に定義されています。しかし AI では、正解が常に明確とは限りません。たとえば、チャットボットにとって「正しい」返答とは何でしょうか？明確な答えがないことは、テストの自動化やカバレッジ追跡を極めて困難にします。

4　AI ソフトウェアには明確な仕様がない

ソフトウェアテスターは通常、システムの動作を記述した仕様書に基づいてテストを行います。しかし、多くの AI システムには明確なルールが存在しません。たとえば、ローン審査モデルには「信用度を予測する」とだけ指示されても、正確なロジックは与えられません。代わりに、モデルは過去のデータからパターンを学習します。

この仕様の欠如により、多くのテストにおいて合格・不合格の判断基準を定義することがほぼ不可能になります。モデルが平均的には優れていても、例外的なケースやレアな入力に対しては誤った、あるいは偏った結果を出すことがあります。明確なルールがない限り、テスターはそのような場合にモデルがどうあるべきかを断言できません。

5　AI の動作はデータに依存する

AI の出力は、コードだけでなく、それが学習したデータによって大きく左右されます。トレーニングデータに偏りや欠損、ノイズがあれば、AI もその問題を反映します。たとえば、顔認識システムが白人の写真ばかりで学習されていれば、黒人の顔に対しては精度が低くなる可能性があります [3]。

このようなデータ依存性により、AI ソフトウェアのテストではコードだけでなく、データパイプライン、データの品質、ラベル付けの精度も含めて検証する必要があります。これは従来のソフトウェアテストよりもはるかに広範で難易度の高い作業です。さらに、テストデータとトレーニングデータの重複が多すぎると、正確な評価ができなくなるおそれがあります。

6　無意識の要求がテストを困難にする

AI システムに対する一部の期待は、明文化されることなく存在しています。これを「無意識の要求」と呼びます。たとえば、ユーザーは AI アシスタントが丁寧な言葉遣いや一般的なアクセントを理解してくれることを当然と考えています。これができないと、仕様書に書かれていなくても「壊れている」と感じられてしまいます。

無意識の要求は特に危険です。というのも、これらはリリース後に発覚することが多く、設計段階で検出されることがほとんどないからです。

このような隠れた期待を発見するには、テスターが経験に基づく手法（探索的テスト、文脈調査、現場観察など）を活用する必要があります。自動運転車やデジタルアシスタントなど、複雑な AI システムでは、こうした無意識の要求が何百項目にも及ぶことがあります。

7　数値の不安定性は発見が難しい

AI システムは多数の数値計算を含みます。特に浮動小数点数を使う計算では、わずかな違いが重大な誤差につながることがあります。これを「数値の不安定性」と呼びます。たとえば、ディープラーニングモデルでは、32 ビットか 16 ビットかで計算結果が変わることがあります。

この問題は発見が難しいのが特徴です。稀な例外ケースや複数の処理を通したあとにしか現れないこともあります。すべての機能テストに合格しているモデルでも、本番環境ではハードウェアやコンパイラ、最適化ライブラリの違いにより不安定な結果を出すことがあります。

8　従来のテストが通用しない理由

従来のソフトウェアテストは主に 2 つのアプローチに基づいています。ホワイトボックステスト（内部コードを調査する）とブラックボックステスト（入力と出力のみを見る）です。AI ソフトウェアはこの両方を打ち破ります。

ホワイトボックステストは、AI システム、特にディープラーニングの内部構造が解釈困難であるため、効果的に機能しません。ニューラルネットワークは読みやすいロジックに従っておらず、その動作は数百万の数値（重み）によって決まります。

ブラックボックステストは、予測可能で決定的な動作を前提としていますが、AI にはその性質がありません。また、AI ソフトウェアの可能な入力（画像、文章、センサーのデータなど）はあまりにも多く、すべてをテストするのは現実的に不可能です [4]。

9　AI ソフトウェアのテスト：新たな技術

このような課題に対応するため、AI 専用の新しいテスト手法が開発されています。

- メタモルフィックテスト：入力と出力の間の期待される関係が保たれているかを確認する方法。たとえば、画像を少し回転させてもラベルは同じであるべきという考え方。正解が分からなくても機能する [5]。
- アドバーサリアルテスト：入力にわずかな変更を加えて、システムが壊れるかどうかを検証する。画像認識や音声認識でよく用いられる。知覚できないノイズでモデルを欺く [6]。
- バイアスと公平性のテスト：異なる人種・性別・年齢などのグループに対する AI の対応を測定する。採用、融資、医療などの分野で特に重要。
- ロバストネスとストレステスト：予期しない状況や過酷な条件下でシステムがどう振る舞うかを確認する。データ欠損、誤入力、低品質なセンサーデータなど。
- シミュレーションテスト：自動運転車のような安全性が重要な領域では、仮想環境で危険な状況（霧、飛び出しなど）を再現し、安全にモデルの反応を検証する。

10　実例：無意識の要求が引き起こす問題

無意識の要求による問題は、ユーザーと直接やり取りする AI システムで特に顕著です。たとえば、AI のカスタマーサポートチャットボットが「怒っている」と言うユーザーに対して、陽気なセールスピッチで返答するようなケースです。技術的にはすべてのテス

トを通過していても、人間の期待には外れてしまいます。

無意識な要求―明文化されていないが、暗黙のうちに期待されているもの―は、システムが導入された後に失敗を引き起こすことがよくあります。これは、誰も正式に書いたわけではないのに、使用者が当然のこととして期待しているものです。仕様書に存在しないため、ほとんどテストされることはありません。しかし、こうした期待が裏切られると、その結果は多くの場合、不満、システムの拒否、そして信頼の喪失につながります。

このギャップを埋めるには、テスターがユーザーの利用状況を観察したり、既存システムの未記載の挙動を調べたり、「弟子入り」や「文脈調査」といった手法を使って、関係者の暗黙の期待を掘り起こす必要があります。

11 新たな QA のあり方へ

こうした課題を受け、AI の品質保証（QA）は新たな考え方を必要としています。モデル、データ、文脈を等しく重視し、仕様には「やるべきこと」だけでなく「やってはいけないこと」も含めるべきです。たとえば、偏見の助長や例外的状況での不安定な動作などを防ぐこと。

AI の QA では、ドメイン専門家、倫理学者、ユーザーを含む多職種のレビューが求められます。テストは一度限りではなく、継続的に行う必要があります。モデルは新たなデータで再学習されることが多いためです。また、トレーニングデータからモデルの意思決定までの追跡可能性（トレーサビリティ）を開発パイプラインに組み込む必要があります。

12 結論：AI のテストは「未知」をテストすること

AI ソフトウェアは強力ですが、予測不可能です。自ら設計したわけではないデータから学び、明確なルールや期待を持たないことも多いのです。従来のテスト手法は通用しません。不確実性、データの偏り、数値の脆弱性、ユーザーの無意識的な期待などを考慮した新たな手法が必要です。

AI ソフトウェアを正しくテストするということは、コードだけでなく、データ、意図、倫理、そして「誰も書いていないが、皆が当然と思っていること」までも検証するということなのです。

参 考 文 献

[1] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, “Machine learning testing: Survey, landscapes and horizons,” *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 1–36, 2020.

[2] E. T. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, “The oracle problem in software testing: A survey,” *IEEE Transactions on Software Engineering*, vol. 41, no. 5, pp. 507–525, 2014.

[3] J. Buolamwini and T. Gebru, “Gender shades: Intersectional accuracy disparities in commercial gender classification,” in *Conference on fairness, accountability and transparency*, pp. 77–91, PMLR, 2018.

[4] V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss, and P. Tonella, “Testing machine learning based systems: a systematic mapping,” *Empirical Software Engineering*, vol. 25, no. 6, pp. 5193–5254, 2020.

[5] F. U. Rehman and M. Srinivasan, “Metamorphic testing for machine learning: Applicability, challenges, and research opportunities,” in *2023 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pp. 34–39, IEEE, 2023.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint arXiv:1412.6572*, 2014.