機械学習適用コンポーネントを含むソフトウェアの形式検証とテスト

來間 啓伸[1]　明神 智之[1]　佐藤 直人[1]　中川 雄一郎[1]　小川 秀人[1]

概要：計算機システムの処理対象が複雑になり，人手のみでの設計・実装が困難になったことから，機械学習を適用した処理コンポーネントの利用が求められている．一方，このようなコンポーネントの動作には本質的な不確かさがあり，実装が仕様を満たすことを確認する，通常のテスト・検証手法は適用できない．ここでは，不確かさを持つコンポーネントを含むソフトウェアの検証とテストにおける，リファインメントの利用について考える．不確かさを非決定性を使って表現することで，不確かさを持つコンポーネントについてはテストのための部分仕様を提供するとともに，そうでないコンポーネントについては形式的な検証が可能になる．

Formal Verification and Testing of Software Including Components Constructed Using Machine Learning

1. はじめに

情報処理装置が発達し，高度なセンサを使った実世界データの収集と様々なアクチュエータを介した実世界への作用が可能になったことで，計算機システムは実世界と密接に結びついて動作するようになった．このようなシステムでは，複雑に関連する大量のデータを処理するソフトウェアを人手で設計し実装することが困難であり，機械学習を適用した処理コンポーネントの利用が必要になる．

機械学習を適用したコンポーネントの動作は，基本的に訓練データセットによって決定されるため，学習アルゴリズムの動作の正しさが検証できても，コンポーネントの動作が正しいとは言えない．このようなコンポーネントの安全性評価における課題として，以下が指摘されている [6]．

(1) 学習を通じてニューラルネットワークに蓄積された知識は人間にとって解釈困難であり，コンポーネントが意図通り動作するかどうかが不透明
(2) コンポーネントの動作に誤差率を含むので，その正しさの評価には統計的なゆらぎがともなう
(3) 訓練データセットは可能な入力データすべてをカバーできない点で不完全であり，コンポーネントの動作には訓練データセットへの依存性がある
(4) 同一の訓練データセットを使っても学習過程によりコンポーネントの動作や蓄積される知識が異なり得るため，安定した評価が困難

これらの不確かさから，実装が仕様を満たすことを確認する通常のテスト・検証手法は有効ではなく，機械学習を適用したコンポーネントのためのテスト・検証手法の研究が進められている．ニューラルネットワークのノードの活性状態を解析し，望ましい性質を満たさない入出力データが存在しないことを検証する研究 [4] などは，内部を解析するホワイトボックス・アプローチの例である．一方，入出力のみに着目するブラックボックス・アプローチの例として，既知の入出力データから入力データを系統的にずらして出力データに期待されるずれが起こることを確認する，メタモルフィック・テスティング [2] があげられる．

一方，機械学習を適用したコンポーネントを含むソフトウェアのテスト・検証においては，対象ソフトウェア全体についての不確かさを前提とする方法は，網羅性や信頼性，効率の点で疑問がある．不確かさを持つコンポーネントとそうでない部分を分離し，前者については不確かさを考慮する一方，後者については従来から確立された方法でテス

[1] 株式会社日立製作所　研究開発グループ　システムイノベーションセンタ
Hitachi, Ltd., Research & Development Group, Yokohama Research Laboratory, 292 Yoshida, Totsuka, Yokohama 244–0817, Japan

ト・検証できることが望ましい．

2. リファインメント

Dijkstra は，プログラムとその正しさの証明を共に成長させる，構成的なアプローチを提案した [1]．構成的なアプローチの実現の一つである B メソッドは，記述の整合性とリファインメントの正しさを数学的基盤のもとに検証する方法を与えており，機能仕様を形式的に記述するとともに，リファインメントにより記述を段階的に詳細化して，最終的にプログラムを導出する．ここで，リファインメントの各段階では上位の記述が下位の記述の動作を全て定める必要はなく，通常は下位の記述の作成者に設計上の選択肢が残される．すなわち，上位の記述はその段階で求められる抽象度でソフトウェアの動作を規定すれば十分であり，より詳細な動作の規定は下位の記述に任せることができる．このような設計上未定であることによる不確かさを残しつつ，各抽象度で記述の整合性を検証可能にするしくみとして，非決定性が使われる．この場合，上位の記述では非決定性が大きく，下位の記述は上位の記述に許容される範囲内で，非決定性が小さいか等しくなければならない．

機械学習を適用したコンポーネントは，リファインメントの開始点になる機能仕様を明確に記述できず，詳細化に相当する学習の過程を厳密に検証する手段もないので，リファインメントを使ったプログラム導出が行えない．一方，機械学習の適用を実装に向けた設計上の選択と位置づければ，このようなコンポーネントを含むソフトウェア全体については，従来通り機能仕様を記述し，リファインメントによってプログラムを導出することが可能と考えられる．このとき，不確かさを含むコンポーネントはリファインメントのどこかの段階でプログラム導出から除外されるが，その動作が非決定的に表現されるので，残りの記述の整合性とリファインメントの正しさは非決定性のもとで検証できる．また，非決定的に表現されたコンポーネントの動作は，ソフトウェアの他の部分とのインタフェースを規定する部分仕様として，コンポーネントのテストで利用できる．

3. 非決定性を使った不確かさの表現

自動車ドアロックシステムを題材とした動作分析のケーススタディ [5] を例に，不確かさを非決定性で表現した記述と検証を示す．このシステムは，一定の時間間隔で車速を測定するセンサ，センサからの車速値と乗員の操作によってドアロック・アンロックを判断するコントローラ，コントローラからの指示にしたがってロック・アンロック動作を行うアクチュエータから構成される．コントローラは車速が定められた値以上のときは乗員のアンロック操作を禁止し，さらに車速が高いときにはアクチュエータにロックを指示する．一方，アクチュエータの動作には遅延があり，動作中にコントローラの指示が変わり得る．車速は，運転者や機械学習を適用した運転支援装置によって，ドアロックシステムとは独立に変化する．すなわち，車速を制御するコンポーネントには不確かさがある．

ドアロックシステムでは，1 センサ周期内に起こり得るセンサ値の最大変化量を考え，センサ値は現在値 ± 最大変化量の間で非決定的に変化するとして車速をモデル化した．このように非決定性を使って表現したモデルの下で，ドアロックシステムに要求される安全要件「車速が一定値以上の時には必ずドアがロックされていること」が証明できる．前提となる「1 周期内の車速変化が最大変化量以下であること」は車速を制御するコンポーネントの部分仕様であり，コンポーネントの不確かさが部分仕様の許容範囲内にあることが求められる．これは，コンポーネントが部分仕様を満たすことのテストによって確認できる．

4. おわりに

機械学習は，従来型の仕様が記述困難な処理で，有用性が高い．ここでは，仕様として記述できない動作を不確かさと位置づけ，不確かさを持つコンポーネントを含むソフトウェアの検証とテストにおける，リファインメントの利用について考えた．コンポーネントの不確かさを非決定的表現を使って記述することで，不確かさを持つコンポーネントについてはテストのための部分仕様を提供するとともに，そうでない部分についてはリファインメントに基づく検証を可能にすることが狙いである．3 節に述べたように，非決定性のもとでも有用な性質を証明することができる．一方，このケーススタディはシステム分析を目的としているため，プログラムの導出は行っていない．不確かさを含まない部分についてのプログラムの導出と，不確かさを含むコンポーネントが導出されたプログラムに整合的に組み込まれることの評価は，今後の課題である．

参考文献

[1] Dijkstra, E.W.: *The Humble Programmer*, CACM, Vol.15, Issue 10, pp.859-866 (1972)
[2] Ding, J., Kang, X., Hu, X-H.: *Validating a Deep Learning Framework by Metamorphic Testing*, 2nd International Workshop on Metamorphic Testing, pp.28-34 (2017).
[3] 石川冬樹, 來間啓伸, 中島震: 不確かさを考慮したソフトウェア・テスティングおよび形式検証, 情報処理, 58(8), pp.693-695 (2017).
[4] Katz, G., Barrett, C., Dill, D. Julian, K. and Kochenderfer, M..: *A Efficient SMT Solver for Verifying Deep Neural Networks*, arXiv: 1702.00135v2 (2017).
[5] 來間啓伸, 中島震: Event-B を使った時間依存性のあるシステムのモデル化と動作分析のケーススタディ, 情報処理学会論文誌, 57(8), pp.1690-1702 (2016).
[6] Salay, R., Queiroz, R. and Czarnecki, K.: *An Analysis of ISO 26262: Using Machine Learning Safely in Automotive Software*, arXiv:1709.02435 (2017).