

## セッション紹介 「機械学習工学—機械学習のためのソフトウェア工学—」

石川冬樹<sup>†1</sup> 今井健男<sup>†2</sup> 丸山宏<sup>†3</sup> 吉岡信和<sup>†1</sup>

**概要:** 近年の深層学習を軸とした機械学習の発展に伴い、機械学習を利用するソフトウェアは急速に社会に浸透しつつある。しかしその一方で、従来型のソフトウェア工学は機械学習の前に全くと言っていいほど通用していない。機械学習ソフトウェアの開発・テスト・運用の方法論は未だに確立できておらず、開発現場では試行錯誤に依っている状況である。この現状を踏まえ、機械学習ソフトウェアに対しては「機械学習工学」ともいうべき、新たなパラダイムの確立・体系化が必要である。以上の認識に立って、本セッションでは、機械学習エンジニア、ソフトウェアエンジニア双方の立場から、機械学習システムに対する問題提起や研究報告、参加者からの取り組み、あるいはこれから取り組んでみたいという思いについて議論したい。

### Machine Learning Systems Engineering — Software Engineering for Machine Learning —

FUYUKI ISHIKAWA<sup>†1</sup> TAKEO IMAI<sup>†2</sup>  
HIROSHI MARUYAMA<sup>†3</sup> NOBUKAZU YOSHIOKA<sup>†1</sup>

**Abstract:** Recent advancement of machine learning, especially deep learning, has brought a large amount of software applying the machine learning technology into our society quite rapidly. On the other hand, however, traditional software engineering techniques has had almost no effect to organize machine-learning based systems; we have failed to establish any methodology for development, testing, deployment and maintenance of machine learning systems, hence the engineers are now falling into trial-and-error in their development. In the light of such current situation, this session would like to present problems, report studies, and discuss the current (or future) effort of attendees, from both of machine learning engineers' and software engineers' perspectives.

#### 1. はじめに

2010年代に入ってから「データサイエンティストブーム」「第3次AIブーム」に伴う機械学習の急速な発展と実用化については、恐らく論を待たないところであろう。1990年代にはサポートベクターマシン (SVM) を中心にした、そして2000年代にはバイズ的手法などを中心とした研究開発は着々となされていたが、2010年以降のビッグデータの勃興とともに、機械学習は「データサイエンス」の関連分野として注目を浴びるようになった。

そして「AIブーム」の火付け役となったのは深層学習 (ディープラーニング) の躍進である。2012年に Hinton 率いるチームが提案する AlexNet[1]が、国際的な画像認識の競技会 ILSVRC で従来手法を10ポイントも改善する精度で優勝したことを契機に、深層学習を応用した画像認識は急速に発展し、2015年には人間の認識精度をも上回ってしまった[2]。深層学習の応用は他分野でも進み、2016年には AlphaGo[3]が囲碁における世界トップレベルのプロ棋士を破って国際的に注目を浴び、人工知能ブームを象徴する出来事となった。

こうして、機械学習・深層学習は世界中でのホットトピ

クとなって実社会に急速に浸透しつつある。2020年には機械学習などを担う「先端IT人材」が国内で4.8万人不足する、という調査もある[4]。

こうした世界的な潮流の陰で、機械学習を用いたシステム (機械学習システム) は様々な面において、実製品・サービスとしてのソフトウェア品質を担保するのが非常に難しいことが、Sculleyら[4]などを端緒として、機械学習のコミュニティでは度々指摘されていた。そして事実、統計的機械学習特有の確率的で不確実な振る舞いは、従来型の要求分析、テスト、開発プロセス、といった様々なソフトウェア工学的手法を、事実上適用不可能にしてしまった。拗って立つ手法が確立しないまま、機械学習システムを扱う開発現場では試行錯誤に依っているのが現状である。

こうした、圧倒的な人材不足と工学的手法の不在という状況は、1960年代の「ソフトウェア危機」を想起させる。これが「ソフトウェア工学」が興る契機となったように、現在、機械学習システムに対して「機械学習工学」ともいうべき、新たなパラダイムの確立・体系化が必要である、と我々は考える。

<sup>†1</sup> 国立情報学研究所  
National Institute of Informatics  
<sup>†2</sup> LeapMind 株式会社  
LeapMind, Inc.

<sup>†3</sup> 株式会社 Preferred Networks  
Preferred Networks, Inc.

## 1.1 「機械学習工学」の研究動向

これまで、機械学習関係のカンファレンス、あるいは関連ワークショップにおいて、いくつかこうした議題を扱う機会があり、それに伴う研究もいくつか発表されている。

翻って、ソフトウェア工学分野においては「ML4SE (ソフトウェア工学のための機械学習適用)」を扱った研究は幾つかあっても、本セッションで取り上げるような「SE4ML (機械学習のためのソフトウェア工学)」といった観点での研究は、僅かな例外を除いてほとんどなされてこなかった。あるいは少なくとも、目に見える形で発表はなされてこなかったのが実状である。

## 1.2 我々の活動の経緯

以前より、丸山による「機械学習工学」の提唱 ([6]など) があり、続いて SES2017[7]でのパネルディスカッション「機械学習とソフトウェア工学」を契機に集った我々は、IT 勉強会形式でのイベント[7]の開催、FOSE2017でのポスター発表[8]を行ってきた。そして本セッションの企画を経て、2018年度、石川を主査として、日本ソフトウェア科学会「機械学習工学研究会」立ち上げに向けて現在活動しているところである。

## 2. 機械学習システムを巡る様々な課題

以降、ソフトウェア工学の中でのそれぞれの分野における主な課題や問題点、そして現在わずかながらにも提案されている解決策を挙げておく。

### 2.1 テスト・検証

確率的に振る舞う機械学習機能に対しては、決定的な動作を前提としたテストオラクルを準備するのが難しい。またテストの成否判定が困難であり、明らかなバグが混入していても、人間には判断がつかない可能性が高い。

また、仮に不具合を同定できたとしても、帰納的に構築される機械学習モジュール内部の挙動は把握・説明するのが難しいため、問題の原因同定・修正も厳しい。

さらに深層学習においては、敵対的サンプル[9]の存在が知られており、人間の目にはわずかな誤差にしか思えない程度のノイズの混入でも機械学習モデルは誤った振る舞いをしてしまう事がある。こうした挙動は人間には極めて把握しづらい。

以上を解決するには、従来のテスト手法をそのまま用いるのは難しい。機械学習システムと従来型ソフトウェアシステムとの間での前提条件の差違、従来型テスト手法の活用可能性を探る必要がある。

また、ニューロンカバレッジテスト[10]や機械学習システム向け形式検証手法[11]、サーチベースドテスト[12]、メタモルフィックテスト[13]など、機械学習システムにより適合したテスト・検証手法はいくつか提案されてきており、その有用性を確かめる必要がある。

また、それらテスト・検証手法によって確かに品質が担

保されると考えるか、その社会的合意形成も必要であろう。

## 2.2 要求工学

機械学習品質の最も標準的な指標は認識・予測の精度であるが、ではどの程度の精度が出れば顧客が満足するのか、予見積もることは難しい。よって機械学習システムにおいては、顧客のビジネス構造を理解した上で、どの程度の精度があるとROIに見合うのかという目的設計が必要である。しかしその目的をどの程度達成できるかという期待値調整は難しく、その為の適切な手法は我々の知る限り存在していない。

また、機械学習の非機能要件を扱うのは、これまで以上に困難になる。機械学習による結果は、前述の通り、その過程がブラックボックスなため説明が難しく、また同じ手法・データ・学習回数であっても再現するとは限らない。そのような中で顧客に対して学習結果のもっともらしさを伝える必要があるが、どうすれば良いのかは現在解がない。

## 2.3 プロジェクトマネジメント

### 2.3.1 プロセス・運用

先に述べた通り、機械学習の品質評価には精度を基準とせざるを得ない面があるが、目標精度を達成するまでには、採用する特徴量の調整 (feature engineering) や、ハイパーパラメータのチューニングを行いながらの長い時間をかけての試行錯誤が必要である。どのような変更を用いれば目標精度を達成できるか予測するのは極めて困難であり、よって開発スケジュールの見積もりは非常に難しい。

試行錯誤を要するという意味では、ウォーターフォールではなく、アジャイルでの開発がより適しているのかもしれない。あるいは DevOps のように、運用時を含めたカスタマイズ・調整により適していると考えられる。

### 2.3.2 プロセス・運用

学習データを供給しながらの機械学習品質の継続的な磨き込みや試験運用を行おうとすると、機械学習の技術以外に、そのシステムを構築・運用するためのソフトウェア開発の知識が必要となる。しかしながら、この両方のスキルセットを持ち合わせる技術者は現在少なく、よって機械学習アルゴリズムの実装者とソフトウェアとしての開発者を別個に配置し、両者がコミュニケーションを取りつつシステムを実現するケースも多い。しかし、このコミュニケーションコストが非常に高く、また両者の保持する知識が異なるため、うまく接続できないケースが多い。

## 2.4 その他

その他にも、様々な課題や問題点が存在する。以下にいくつかを列挙する。

**学習データと前処理** 学習データと学習データの不足や欠損、あるいは前処理にかかる冗長で泥臭い作業の困難さは、恐らくどの開発現場でも問題になっているところであろう。

**再利用** 学習には多量の計算リソースや工数を要するた

め、学習結果のモデルを再利用することで、より少ない工数での学習精度の担保を行う技術がいくつか提案されている。これらをうまく活用する方法論が確立されるべきであろう。

**説明可能性** 既に何度か述べているが、学習モデルがブラックボックスであり、学習結果の根拠を示せないことは、利用者にとっては機械学習システムへの信頼を妨げる要因となる。

**バージョン管理・構成管理** データから帰納的に構成する機械学習モデルを再現可能にするには、コードのバージョン管理だけではなく、データのバージョン管理も必要となるが、そのためのツールや方法論は普及していない。

**アーキテクチャパターン** 機械学習システムは単独で使われるわけではなく、従来型ソフトウェアシステムと統合された形で使われるはずである。例えば[14]は、B-Treeのインデックス機能を深層学習による予測モデルで置き換えたものであるが、これは従来型のインデックス機能に深層学習機能が統合されたものである。システムのどの機能を機械学習で実現するか、どのように従来型システムと統合するかにおいては何かしらの典型的なアーキテクチャが存在するはずであり、それらをパターンとして明示化することで、開発効率を上げられる可能性がある。

### 3. おわりに

機械学習システムのための工学は、現在は必要性が認識されたのみで、各手法の研究はまだ「これから」の段階にある。本セッションでの議論を通じて今後取り組むべき重要な課題を明らかにし、今後より多くの研究者、あるいは技術者の取組みにつなげられることを期待する。

### 参考文献

- [1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (pp. 1097–1105). Curran Associates Inc. 2012.
- [2] He, K., Zhang, X., Ren, S., & Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Retrieved from <http://arxiv.org/abs/1502.01852>, 2015.
- [3] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... Hassabis, D. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484–489. <https://doi.org/10.1038>, 2016.
- [4] 経済産業省 商務情報政策局 情報処理振興課. IT人材の最新動向と将来推計に関する調査結果 ～ 報告書概要版 ～. 2016.
- [5] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M. Machine Learning: The High Interest Credit Card of Technical Debt. SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop), 2014.
- [6] 丸山 宏, ソフトウェアの役割, コンピュータ ソフトウェア, vol.33, no.4, 2016.
- [7] 情報処理学会 ソフトウェア工学研究会, ソフトウェアエンジニアリングシンポジウム 2017, 2017.
- [8] 吉岡信和, 石川冬樹, 今井健男, 丸山宏. SE チャレンジ: 機械学習 x ソフトウェア工学 = 機械学習工学. 第 24 回ソフトウ

- エア工学の基礎ワークショップ (FOSE2017), 2017.
- [9] He, K., Zhang, X., Ren, S., & Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Retrieved from <http://arxiv.org/abs/1502.01852>, 2015.
  - [10] Pei, K., Cao, Y., Yang, J., & Jana, S. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. <https://doi.org/10.1145/3132747.3132785>, 2017.
  - [11] Huang, X., Kwiatkowska, M., Wang, S., & Wu, M. Safety Verification of Deep Neural Networks. Retrieved from <http://arxiv.org/abs/1610.06940>, 2016.
  - [12] McMinn, P. Search-Based Software Testing: Past, Present and Future. In 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops (pp. 153–163). IEEE. 2011.
  - [13] Segura, S., Fraser, G., Sanchez, A. B., & Ruiz-Cortes, A. A Survey on Metamorphic Testing. IEEE Transactions on Software Engineering, 42(9), 805–824, 2016.
  - [14] Kraska, T., Beutel, A., Chi, E. H., Dean, J., & Polyzotis, N. The Case for Learned Index Structures. Retrieved from <http://arxiv.org/abs/1712.01208>, 2017.