機械学習ソフトウェアのエンジニアリングでの課題

中本 幸一[1,a)]

概要：機械学習ソフトウェア，特に深層学習ソフトウェアを開発する上でのエンジニアリング上の課題を述べる．学習データの評価手法，アタックの脆弱性，ホワイトボックス解析技術の必要性，深層ネットワークの構成の妥当性の確認方法，フレームワークの乱立，深層ネットワーク中間表現の提案と乱立，開発支援システムの必要性である．

1. はじめに

筆者はもともと組込みシステム，特にシステムソフトウェアを専門領域としている．ここ数年，国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務である “IoT 推進のための横断技術開発プロジェクト/省電力 AI エンジンと異種エンジン統合クラウドによる人工知能プラットフォーム”に携わっている．筆者はこの中で機械学習データを入力して推測を高速に行う FPGA 開発フレームワークの研究開発プロジェクト (N3 プロジェクト) に参加している [1]．この開発にあたり，実際の機械学習のプログラム (深層学習プログラム) を読み (読むものではないが)，開発し，評価するようになってから，既存のシステムソフトウェアやソフトウェア工学と異なった問題意識を持つようになった．本稿ではその問題意識を述べる．また，著者が専門とするシステムウェア領域から特に高速に処理を行う観点から研究が行われている．著者が参加した SOSP17(Symposium on Operating Systems Principles)) とそのワークショップ AISys17(Workshop on AI Systems at Symposium on SOSP) [*1] での議論の内容も関連して紹介する．機械学習は対象問題とアルゴリズムにより多くの技術があるが，本稿では深層学習に限定する．

2. ソフトウェア工学の視点から

学習データの重要性: 深層学習ソフトウェアは，ニューラルネットの層を設計し，それに学習データを入力させて，ネットを学習させる．この時，学習データは推論を行う決定論理を左右する位置づけである．それであるにも関わらず，学習データの扱いが問題である．例えば，以下のような問題がある．

- 学習データの偏り: 学習データに偏りがあって，その学習データに大きく依存してしまうと過学習 (overfitting) になる [2][Sec. 5.2]．これが起こると新たなデータに対して対応できなくなり，深層学習ソフトウェアでは大きな問題となる．ドロップアウトやネットワークのニューロンへの重みの制限するなどの各種技法が考案されている．
- データ拡張の誤り: 学習データが不足する場合に深層学習ソフトウェアではデータ拡張 (data augmentation) を行われる [2][Sec. 7.4]．これには既にある学習データを回転，移動などの変換をかけて新たに学習データを生成するものである．しかし，この変換を誤ると誤った学習データを入力させることになる．
- ラベリングの間違い: 学習データがどういうデータかの情報を付与することをラベリングといい教師あり学習では必須ものである．学習データが膨大になるとラベリングを誤る可能性が高まる．ラベリングを誤ると，これも誤った学習データを入力させることになる．また，ラベリング作業は膨大な時間を要する場合がある．
- 悪意による学習データ改竄: 上述は学習データを誤るケースだが，一方で悪意で改竄した学習データを混入させることが容易であるとされている [3].

学習データの評価するのに結局学習させてみないと分からないというと何をやっているのか分からなくなる．学習させないで学習データの良し悪しを調べる方法はないものだろうか．

テスト手法: DNN をブラックボックスとして扱い，その認識結果のみを評価するような場合，上述の意図的に改竄された入力データの検出は難しいことが指摘されている [4]．DNN のロジックを扱うホワイトボックス的なアプローチ

[1] 兵庫県立大学
University of Hyogo, Kobe, Hyogo 650–0047, Japan
[a)] nakamoto@ai.u-hyogo.ac.jp
[*1] https://www.sigops.org/sosp/sosp17/, http://learningsys.org/sosp17/

が必要である．

DeepExplore は一つの解を提供してくれるかもしれない [5]．DeepExplore は学習データに対するラベリングの面倒さとニューラルネットワークでのニューロンのカバレッジの低さという問題を解決することを目的とする．まず，ニューラルネットワークでのニューロンのカバレッジを定義する．次にラベル付けされていない学習データを複数のニューラルネットワークに対して学習データを変更して認識結果が異なるようなものを生成する．最後に，ニューロンのカバレッジの向上と認識結果が異なるという 2 つの目標を joint optimization として解き，最終的な学習データを求めている．

ネットワークの構成の妥当性: 畳み込みニューラルネットワーク (CNN) では，畳み込み層やプーリング層などの各種のネットワーク層が多層に重畳された構成をとっているが，なぜそういう構成をとれば認識率が上がるのか，著者にはよく分からない．一方で，プーリングが情報を落としているという反省から，CNN の発明者から Capsule という概念が新たに提案されてきている [6]．ネットワークの構成する層の効果やネットワークの構成の妥当性を測る尺度が必要と考える．

3. システムアーキテクチャの課題

本節では，AIsys17 での発表を中心にシステムアーキテクチャの視点からの 課題を述べる．

深層学習フレームワークの乱立: 深層学習のフレームワークは一般のソフトウェアの世界ではプログラミング言語に相当するものであろう．これが，深層学習の世界では，Caffe[*2], TensorFLow[*3], Chainar[*4], CNTK[*5], Caffe2[*6] と多数あり乱立状態にある．各々既存のフレームワークでは不十分なので新規に開発している，あるいは解く問題用にフレームワークを開発している (一般のソフトウェアの世界での問題向けプログラミング言語) という状況であるかもしれない．利用者から見れば選択に迷う状況である．

統一中間表現の提案とその乱立: 深層学習フレームワーク毎に学習結果を含む各種の中間表現が異なる．著者が関わる N3 プロジェクトでは当面 Caffe の学習結果を FPGA に変換するフレームワークを開発しているが，他の学習フレームワークへの対応を要請される．学習結果の形式が統一されるのが望ましい．AISys17 の発表では，偶然かもしれないが，このような状況を解決するために，幾つかの企業や大学が，複数の深層学習フレームワークに対応した中間表現の提案の発表があった．University of Washington の Tianqi Chen による TVM[7]，Stanford Univ の Matei Zaharia による Weld[8]，Facebook の Yangqing Jia による ONNX である．各々中間表現と言っても力点が違っているようであるまた産業界では，KHRONOS Group が Neural Network Exchange Format を提案している．またも乱立の様相の感は否めない．

開発支援システム: NVIDIA 社を始めとして深層学習関連の製品を出している企業は開発支援ツールを提供している．NVIDIA 社では DIGITS[*7] がある．ただ，本質的なモデル開発支援を含めて深層学習ソフトウェアを容易に開発可能とするためのシステム開発支援が提案されている．Stanford の DAWN[10]，OSU の EasyML[11] などがある．

4. おわりに

最近，機械学習ソフトウェアの研究開発に携わって持った問題意識を述べた．何分，携わって短期なので，思い違いもあるかもしれない．ご指摘頂ければ幸いである．

謝辞

この成果は，国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務の結果得られたものである．

参考文献

[1] 山本他：FPGA を用いた組込みシステム向け深層学習フレームワークの構想, 組込みシステムワークショップ 2017 (2017).
[2] Goodfellow, I., et al.: *Deep Learning*, The MIT Press (2016).
[3] Carlini, N., et al.: Towards Evaluating the Robustness of Neural Networks, *Proc. IEEE Symposium on Security and Privacy*, pp. 39–57 (2017).
[4] Goodfellow, I., et al.: The challenge of verification and testing of machine learning (2017). available from ⟨http://www.cleverhans.io/security/privacy/ml/2017/06/14/verification.html⟩
[5] Pei, K., et al.: DeepXplore: Automated Whitebox Testing of Deep Learning Systems, *Proc. 26th Symposium on Operating Systems Principles*, pp. 1–18 (2017).
[6] Sabour, S., et al.: Dynamic Routing between Capsules, *Advances in Neural Information Processing Systems* (2017).
[7] Chen, T., et al.: TVM: An End to End IR Stack for Deploying Deep Learning Workloads on Hardware Platforms (2017). available from ⟨http://tvmlang.org/2017/08/17/tvm-release-announcement.html⟩
[8] Palkar, S., et al.: Weld: A common runtime for high performance data analytics, *Proc. 8th Birnnial Conf. on Innovative Data Systems Research* (2017).
[9] Khronos Group: Neural Network Exchange Format (2017). available from ⟨https://www.khronos.org/nnef⟩
[10] Bailis, P., et al.: Stanford DAWN (2017). available from ⟨http://dawn.cs.stanford.edu/⟩
[11] Hendricks, P., et al.: Easyml: Easily Build And Evaluate Machine Learning Models, *bioRxiv* (2017).

[*2] http://caffe.berkeleyvision.org/
[*3] https://www.tensorflow.org/
[*4] https://github.com/chainer/chainer
[*5] https://www.microsoft.com/en-us/cognitive-toolkit/
[*6] https://research.fb.com/downloads/caffe2/
[*7] https://developer.nvidia.com/digits