

Towards Training Reproducible Deep Learning Models

[ICSE 2022](#) [リンク](#)

Boyuan Chen (Centre for Software Excellence, Huawei Canada Kingston, Canada),

Mingzhi Wen (Huawei Technologies Shenzhen, China),

Yong Shi (Huawei Technologies Shenzhen, China),

Dayi Lin (Centre for Software Excellence, Huawei Canada Kingston, Canada),

Gopi Krishnan Rajbahadur (Centre for Software Excellence, Huawei Canada Kingston, Canada),

Zhen Ming (Jack) Jiang (York University Toronto, Canada)

発表者: 田口 舞奈

2022/9/26

人工知能の急速な発展

■ 研究・実践での急速な発展

- AIベースアプリの潜在的市場価値：年間3.5~5.8兆ドル^[11]
- DLモデルで実現：自動運転^[34]，音声認識^[24]，ヘルスケア^[29]

■ 深層学習(DL)における再現性

- 製品開発やオープンサイエンスな研究動向の促進

△ 最近の研究は再現性が保たれていない

- AIの再現性^[37,41]
- DLモデルの再現性^[32, 44, 48, 50, 56, 58, 63, 65]

[11] : 2021 (accessed August, 2021). Notes from the AI Frontier Insights from Hundreds of Use Cases. <https://www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-applications-and-value-of-deep-learning>

[34] :] Sorin Mihai Grigorescu, Bogdan Trasnea, Tiberiu T. Cocias, and Gigel Macesanu. 2020. A survey of deep learning techniques for autonomous driving. J. Field Robotics 37, 3 (2020), 362–386.

[24] : Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse H. Engel, Linxi Fan, Christopher Fougner, Awni Y. Hannun, Billy Jun, Tony Han, Patrick LeGresley, Xiangang Li, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Sheng Qian, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Chong Wang, Yi Wang, Zhiqian Wang, Bo Xiao, Yan Xie, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. 2016. Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin. In Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19–24, 2016 (JMLR Workshop and Conference Proceedings).

[29] : Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. 2019. A guide to deep learning in healthcare. Nature medicine 25, 1 (2019), 24–29.

[37] : Odd Erik Gundersen and Sigbjørn Kjensmo. [n.d.]. State of the Art: Reproducibility in Artificial Intelligence. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18).

[41] : Matthew Hutson. 2018. Artificial intelligence faces reproducibility crisis. Science (New York, N.Y.) 359 (02 2018), 725–726. <https://doi.org/10.1126/science.359.6377.725>

[32–65]は最終ページに

深層学習における再現性の課題

➤ ソフトウェアのランダム性

- データシャッフル, 重み初期化等 ⇒ モデル構築に必須^[14,56]
- ランダムシード ⇒ 最適値の設定が難しい

➤ ハードウェアの非決定性

- 膨大な浮動小数点演算の丸め誤差
- GPUによるアルゴリズムの自動チューニング機能

➤ 資産管理の体系的なガイドラインの欠如

- 一般のソースコードと異なるツール(Git以外)での管理が必要^[23]

[14] : 2021 (accessed August, 2021). Reproducibility in Pytorch. <https://pytorch.org/docs/stable/notes/randomness.html>

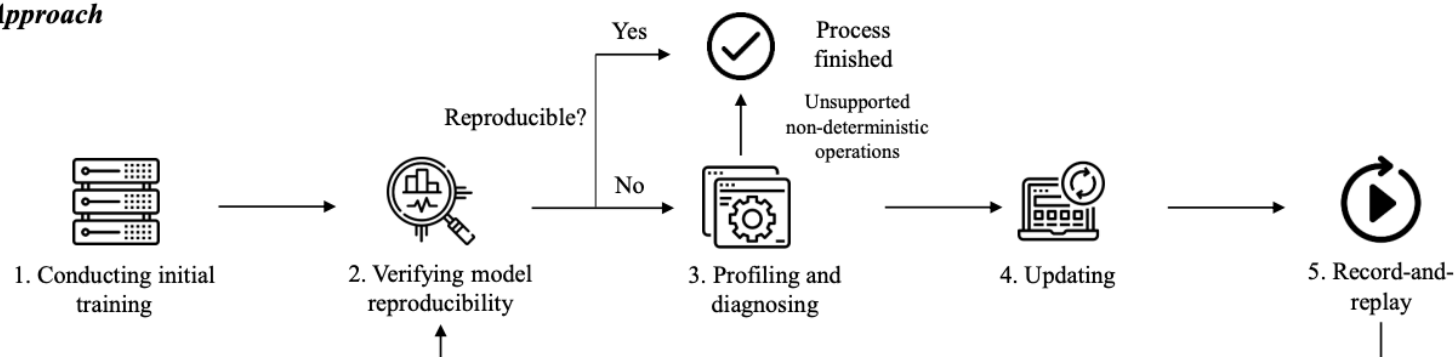
[56] : Hung Viet Pham, Shangshu Qian, Jiannan Wang, Thibaud Lutellier, Jonathan Rosenthal, Lin Tan, Yaoliang Yu, and Nachiappan Nagappan. [n.d.]. Problems and Opportunities in Training Deep Learning Software Systems: An Analysis of Variance. In 35th IEEE/ACM International Conference on Automated Software Engineering, ASE 2020, Melbourne, Australia, September 21-25, 2020

[23] : Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald C. Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software engineering for machine learning: a case study. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE (SEIP) 2019, Montreal, QC, Canada, May 25-31, 2019*, Helen Sharp and Mike Whalen (Eds.). IEEE / ACM, 291-300.

目的とアプローチの概要

- 再現可能な深層学習モデルを構築するための体系的なアプローチの提案

Approach



Artifacts

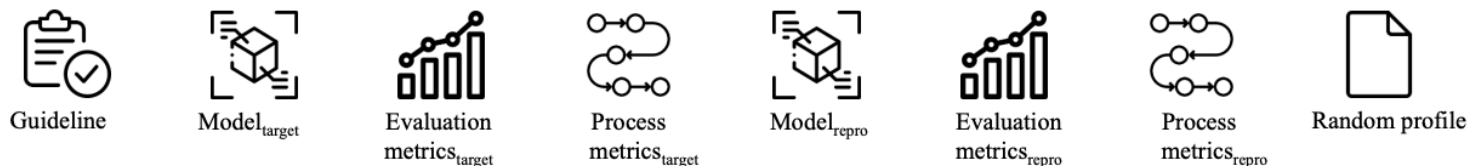


Figure 1: An overview of our approach.

Phase1：初期トレーニングの実施

1. 実験モデルの構築

※SUSE Linux Enterprise Server 12,
Tesla-P100-16GB GPU ver1.14,
CUDA 10.0.130とCuDNN 7.5.1

勾配法の手描き文字認識モデル[47,56]

Python3.6

Tensor Flow

2. 対象DLモデルの学習

- LeNet-5学習スクリプトを用いて
損失値, 学習エポック, 学習時間の
3つのプロセス指標を収集[56]

トレーニング
画像 10,000枚

検証
7,500枚

テスト
2,500枚

3. 対象モデルの検証と再現モデルの構築

- 対象モデルが1.の情報と一致しているか検証
- 2.と同じ手順で再現モデルを生成

[56] : Hung Viet Pham, Shangshu Qian, Jiannan Wang, Thibaud Lutellier, Jonathan Rosenthal, Lin Tan, Yaoliang Yu, and Nachiappan Nagappan. [n.d.]. Problems and Opportunities in Training Deep Learning Software Systems: An Analysis of Variance. In 35th IEEE/ACM International Conference on Automated Software Engineering, ASE 2020, Melbourne, Australia, September 21-25, 2020.

Phase2：モデルの再現性の検証

1. タスクごとの評価指標による学習結果の検証

- 分類タスク：総合精度, クラスごとの精度, テストデータの予測結果
- 回帰タスク：分類タスクの平均絶対誤差(評価指標)

2. 対象/再現モデルから収集した指標の検証

- プロセス指標(エポック数と各エポックの損失値)
- 評価指標 ⇐ それぞれ完全に一致するかどうか

3. 結果の報告

- 2つの指標が完全に一致していれば再現可能である
- 再現可能でなければ次のフェーズへ

Phase3：ランダム性とその原因の特定

1. ランダム性の特定

- OSカーネル呼び出し ⇒ straceで学習プロセスとのやりとりを列挙
- ライブラリ呼び出し ⇒ cProfileでモデルが呼び出す関数を列挙

2. ランダム性の原因の特定

- ソフトウェア関連の原因
- ハードウェア関連の原因

```
Line System calls
10  execve("/usr/bin/python3", ["python3", "mnist_lenet_5.py"],
    0x7fffd6d52180 /* 25 vars */) = 0
...
20  openat(AT_FDCWD, "/dev/urandom", O_RDONLY) = 4
...
51  getrandom("\xf6\xc3\x44\xe2\x06\x65\x95\x70\xca\x48\x4b\xd3\x65
    \x9d\xcb\x8f", 16, 0) = 16
...
100 exit_group(0) = ?
101 +++ exited with 0 +++
```

(a). The sample outputs of strace.

```
ncalls  Filename:lineno(function)
3       nn_ops.py:2876 (tf.nn.softmax)
101     base_layer.py:742 (weights)
...
2       nn_ops.py:2627 (tf.nn.bias_add)
```

(b). The sample outputs of cProfile.

- **bias_add**は、浮動小数点の丸め誤差を引き起こす
並列計算の順番が未確定なatomicAdd()の
CUDA実装を利用している

Phase4：アップデート

1. 記録用OSカーネル呼び出しのリストの更新

- ランダム性を導入した関数をOSカーネル呼び出しのリストに全て記録する

(例)：**read/dev/urandom, getrandom**

2. ランダム性に関わるライブラリ呼び出しに適切なパッチを適用

(例)：**bias_add**を**_patch_bias_add**に置き換える

Phase5：記録と再生

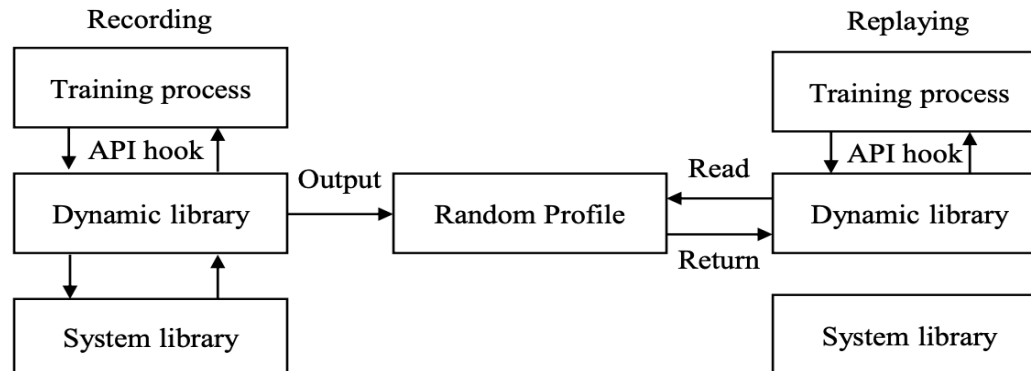


Figure 3: Our record-and-replay technique.

1. 記録

- 対象モデル学習中に、ランダム性に関わるOSカーネル呼び出しごとに戻り値を「ランダムプロファイル」に保存

2. 再生

- 「ランダムプロファイル」に格納されたランダム値を使って再現モデルを学習する

実験モデルとその設定一覧

対象

- 先行研究[35, 38, 49, 51, 52, 56]と同様に画像処理に関する6つの人気なオープンソースDLモデル
- Huawei社の商用システムモデルX

Table 1: The DL models used in our case study.

Models	Datasets	# of Labels	Setup	Task
LeNet-1 [47] LeNet-4 [47] LeNet-5 [47]	MNIST [9]	10	All	Classification
ResNet-38 [39] ResNet-56 [39]	CIFAR-10 [3]	10		
WRN-28-10 [69]	CIFAR-100 [3]	100	G1-G10	
ModelX	Dataset X	-	G1-G5	Regression

Table 2: The information of all the experiment setups. R&R represents Record-and-Replay.

ID	Hardware	Software	Seed	R & R	Patch
C1	CPU	TF1.14	-	-	-
C2			Yes	-	-
C3			-	Yes	-
C4		TF2.1	-	-	-
C5			Yes	-	-
C6			-	Yes	-
G1	GPU	TF1.14	-	-	-
G2			Yes	-	-
G3			-	-	Yes
G4			Yes	-	Yes
G5			-	Yes	Yes
G6		TF2.1	-	-	-
G7			Yes	-	-
G8			-	-	Yes
G9			Yes	-	Yes
G10			-	Yes	Yes

結果

- **6つ全てのDLモデルを、デフォルト設定での再現に成功**

- ModelXはライブラリ関数を追加のパッチ^[15]で再現

- **時間的オーバーヘッド**

- 提案手法の方が学習時間が長くなっている

⇒G1とG3, G3とG5の設定で統計的有意性を比較すると、ランダムプロファイルの置き換え処理が原因

Table 3: Comparing the time and storage overhead. Time(O) represents the average training time (in hours) for original setup, and Time(R) represents the average training time (in hours) for the setup using our approach (Time(R)). The time is italicized if p-value is <0.001 and the effect size is large with (*). RP represents for Random Profile.

Model	Time(O)/Time(R)	Model Size	RP Size
LeNet-1	<i>0.017/0.023</i> (*)	35 KB	13 KB
LeNet-4	<i>0.019/0.027</i> (*)	224 KB	13 KB
LeNet-5	<i>0.021/0.028</i> (*)	267 KB	13 KB
ResNet-38	<i>1.243/1.561</i> (*)	4.8 MB	13 KB
ResNet-56	<i>1.752/2.179</i> (*)	7.6 MB	13 KB
WRN-28-10	<i>7.08/14.979</i> (*)	279 MB	13 KB
ModelX	-	675 KB	38 KB

議論

- 他のDLフレームワークにおける汎用性：
TensorFlowと新たにKerasを比較している(省略)
- 外部妥当性：他のOSやDLフレームワーク，DLの活用分野についても今後調査予定
- 内部妥当性：パッチ適用/無効にした実行のみで比較し，同じ依存性が確認された
- 構成妥当性：この実験で利用したDLモデルは，慎重なレビューが行われている[35, 38, 52, 56]

結論

- 深層学習の再現可能性への関心が高まっているが、個別のランダム性の原因緩和しか研究されなかった

⇒ソフトウェアとハードウェアのランダム性を制御してDLモデルを再現するアプローチを提案

■ 結果

- OSのDLモデル6つと商用DLモデル1つで再現成功

所感

選んだ理由

- 最近絵を生成する Novel AIなどが注目されている
- 自分の研究で必要なものと似ていると思った

所感

- ランダムシードの欠点が補完されている手法かどうか疑問がいくつか残った