

第36回日本ソフトウェア科学会 大会併設チュートリアル

「深層学習のテスト」 イントロダクション

国立情報学研究所 石川 冬樹

f-ishikawa@nii.ac.jp / @fyufyu

<http://research.nii.ac.jp/~f-ishikawa/>

■ 日本ソフトウェア科学会 機械学習工学研究会

- 2018年4月 正式立ち上げ
- 研究会なので、基本は「場」の提供に専念



■ 先日人工知能学会全国大会にて19回目のイベント

- 他イベント内のセッションも多い（分野横断的）
- 参加者は企業の方々が中心
(学界からの講演会や国際会議の輪講でも)

- 10/18 国際シンポジウム@一橋講堂

■ まだまだこれから

- 学界からの貢献、数理や機械学習の知識に基づいた取り組み強化が必要

皆様もぜひご参加を！

■ QA4AI：AIプロダクト品質保証コンソーシアム

- これも2018年4月発足
- AIプロダクトの品質保証に関する調査・体系化、
適用支援・応用、研究開発、社会の啓発を推進
- 現在 39名 + 3団体

参加&貢献大歓迎

149ページ！



- 2019年5月17日ガイドライン初版リリース
+ 5団体での合同シンポ（今後連携！）
- 今後：他の取り組みと連携しつつ継続更新
 - 例：AIST+NIIによる品質標準の策定
(機能安全およびCommon Criteria2つのアプローチ)

その他

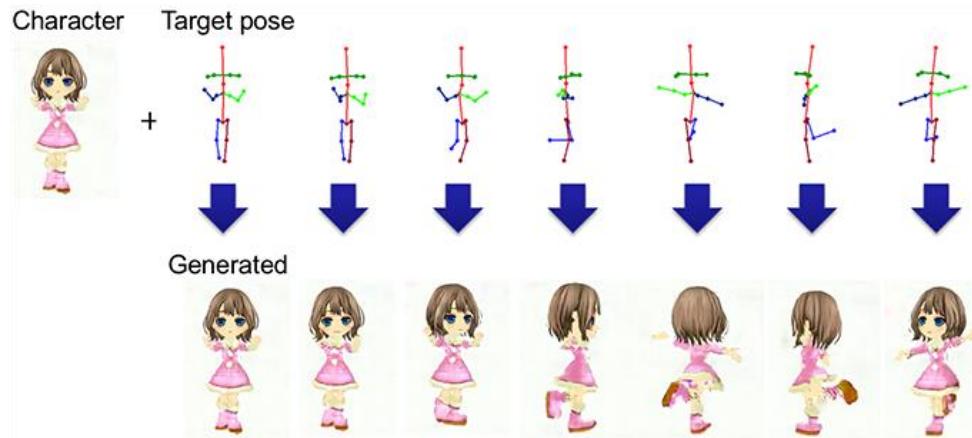
- JST 戦略プロポーザル 報告書
 - 「AI応用システムの安全性・信頼性を確保する新世代ソフトウェア工学の確立」
<https://www.jst.go.jp/crds/report/report01/CRDS-FY2018-SP-03.html>
- NEDOでの公募（今年度は終了）
 - 「次世代人工知能・ロボット中核技術開発」（人工知能の信頼性に関する技術開発）」
- 国内・海外のアカデミック or 産業界向け会議双方において多数の招待講演やパネル討論
- などなど

機械学習応用システムの 品質に関する話題

技術の進化（ごくごく一例）

■DeNAさん（2018年5月）

■画像に対し指定した姿勢の画像を生成可能



画像元：[<https://dena.com/intl/anime-generation/images/anime2.png>]

■動画を生成することもでき、「始点」から「終点」へ連続的に「変身」させることも（服を変えるなど）

ところで皆さんなら何をどう保証（テスト）しますか？

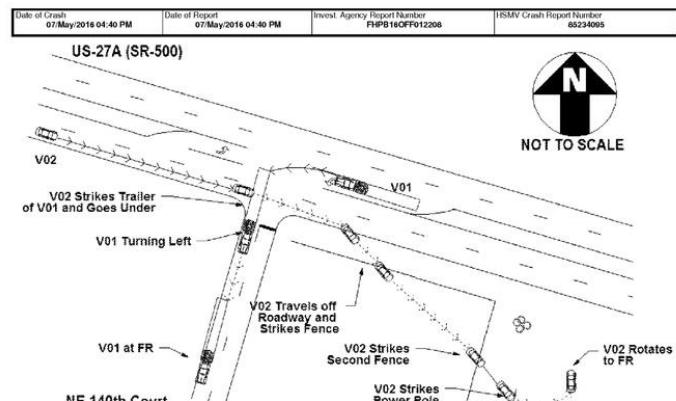
[<https://dena.com/intl/anime-generation/>]

社会的影響の大きい例・使い方を問う例（1）

- 2016年のテスラ自動運転車の事故 ここが機械学習
 - テスラの発表：「まぶしい空に対してトレーラーの白い側面を認識できず」（「運転者すら」ともある）
[<https://www.tesla.com/jp/blog/tragic-loss>]
 - 2017年11月の調査報告での論点に上記は全くなく、
「そもそも自動運転の対象外状況だが運転者が長らく操作していない」という過信や警告音の効力が中心
[<https://dms.ntsb.gov/pubdms/search/hitlist.cfm?docketID=59989>]



[<http://www.dailymail.co.uk/news/article-3677101/Tesla-told-regulators-fatal-Autopilot-crash-nine-days-happened.html>]



社会的影響の大きい例・使い方を問う例（2）

■ 2018年3月・Uberの自動運転車（試験運転中）による死亡事故

■ 最近の事故なので、正式な調査詳細を得ていません

■ 暗がりから突然道路を横断する歩行者

■ 警察官：「人間の運転手だろうが、これは避けられないでの
は」（おそらくレーダーを知らない発言）

[<https://www.recode.net/2018/3/21/17149428/uber-self-driving-fatal-accident-video-tempe-arizona>]
車載カメラの映像がそのまま出ているので要注意

■ 「人かもしれない」と画像認識されたものに対する
ソフトウェアの誤判断があつたらしい

[<https://gigazine.net/news/20180508-uber-fatal-crush-software-bug/>]

■ 緊急ブレーキが無効になっていたらしい

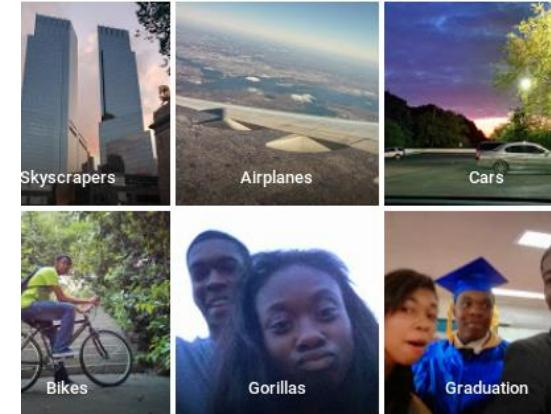
[<https://techcrunch.com/2018/05/24/uber-in-fatal-crash-detected-pedestrian-but-had-emergency-braking-disabled/>]

社会的影響の大きな例・技術的限界の例

■ Googleフォトの画像認識

- 被写体の自動タグ付け機能
- 黒人を「ゴリラ」とタグ付け
- 2年経って本質的には直せていない

(ゴリラを禁止ワード扱い
にして対策)

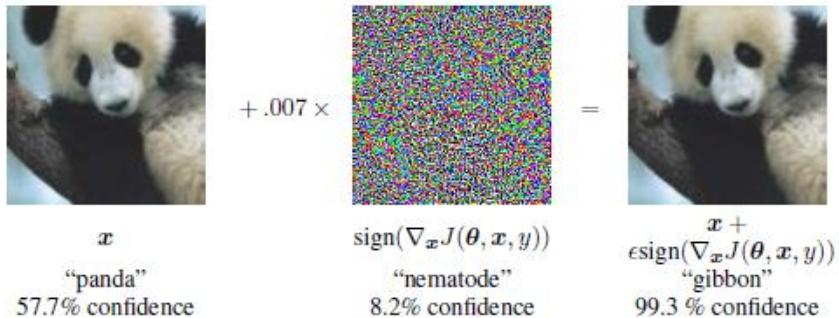


[<https://www.theguardian.com/technology/2015/jul/01/google-sorry-racist-auto-tag-photo-app>]

[<https://www.theguardian.com/technology/2018/jan/12/google-racism-ban-gorilla-black-people>]

よく知られた課題（の一つ）：敵対的サンプル

■ 優れた画像識別器が少しのノイズで誤認識



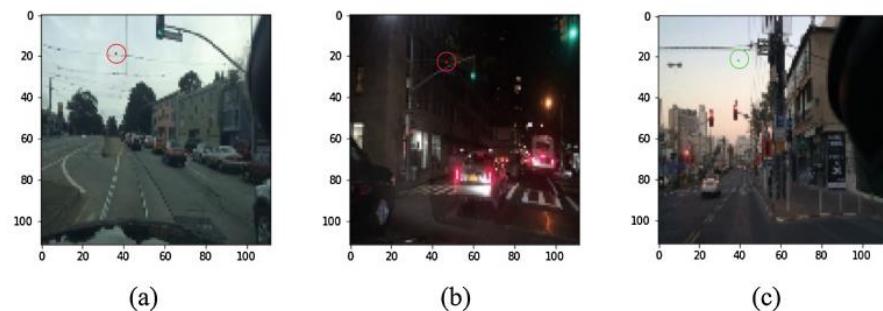
有名な例
「パンダ」が「テナガザル」に

[画像元：Goodfellow et al., Explaining and Harnessing Adversarial Examples, 2015]



物理的なテープ貼付などによる誤認識発生

[画像元：Ackerman, Slight Street Sign Modifications Can Completely Fool Machine Learning Algorithms, 2017]



1ピクセルの変化で信号色の誤認識発生

[画像元：Wicker et al., Feature-Guided Black-Box Safety Testing of Deep Neural Networks, 2018]

訓練データ・ある種の攻撃に関する例

■Twitter Botによる不適切発言

■差別や放送禁止用語を「教えた」ユーザがいた

If you guessed, “It will probably become really racist,” you’ve clearly spent time on the Internet. Less than 24 hours after the bot, [@TayandYou](#), went online Wednesday, Microsoft halted posting from the account and deleted several of its most obscene statements.

The bot, developed by Microsoft’s technology and research and Bing teams, got major assistance in being offensive from users who egged it on. It disputed the existence of the Holocaust, referred to women and minorities with unpublishable words and advocated [genocide](#). Several of the tweets were sent after users commanded the bot to [repeat their own statements](#), and the bot dutifully obliged.

TECHNOLOGY

Microsoft Created a Twitter Bot to Learn From Users. It Quickly Became a Racist Jerk.

By DANIEL VICTOR MARCH 24, 2016



TECHNOLOGY | Microsoft Created a Twitter Bot to Learn From Users. It Quickly Became a Racist Jerk.

Tay's Twitter account. The bot was developed by Microsoft's technology and research and Bing teams.

[<https://www.nytimes.com/2016/03/25/technology/microsoft-created-a-twitter-bot-to-learn-from-users-it-quickly-became-a-racist-jerk.html>]

バイアス・倫理的要求に関する例

■ Amazonの「AI採用」が男女差別をしていた？

■ 訓練データが男性多数

[<https://jp.reuters.com/article/amazon-jobs-ai-analysis-idJPKCN1ML0DN>]

The screenshot shows a news article from Reuters Japan. At the top, there is a navigation bar with the Reuters logo and links to various sections like World, Business, Market, Foreign Exchange, and Video. Below the navigation bar, there is a horizontal menu with categories such as Foreign Exchange, Stock Market, Foreign Exchange Forum, Trade War, North Korea, Trump Policy, Opinion, Economics-Policy, and Technology. The main headline is "焦点：AmazonがAI採用打ち切り、「女性差別」の欠陥露呈で" (Focus: Amazon cuts off AI hiring, reveals defects in 'gender discrimination'). The author is listed as Jeffrey Dastin. There are social sharing icons for Twitter and Facebook. The text of the article discusses how Amazon's AI hiring system discriminated against women due to training data being mostly male.

■ 簡単な計算：

男性90人と女性10人（計100人）に関する推論

男性90%正解, 女性10%正解 → 82/100点

男性80%正解, 女性80%正解 → 80/100点

学んだ傾向・規則の妥当性に関する例

■ 入力画像のどこが結果に効いたか分析してみた

- 「雪」が写っているときは
「オオカミ」であるという
規則を学んでしまった

[Ribeiro et. al., "Why Should I Trust You?":
Explaining the Predictions of Any Classifier, KDD'16]

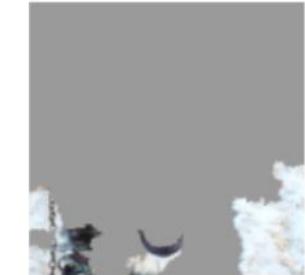
■ どの訓練画像が参考に なったか分析してみた

- 悪い識別器では、「魚」と
いう結果は合っていても
雑な色感しかみてない

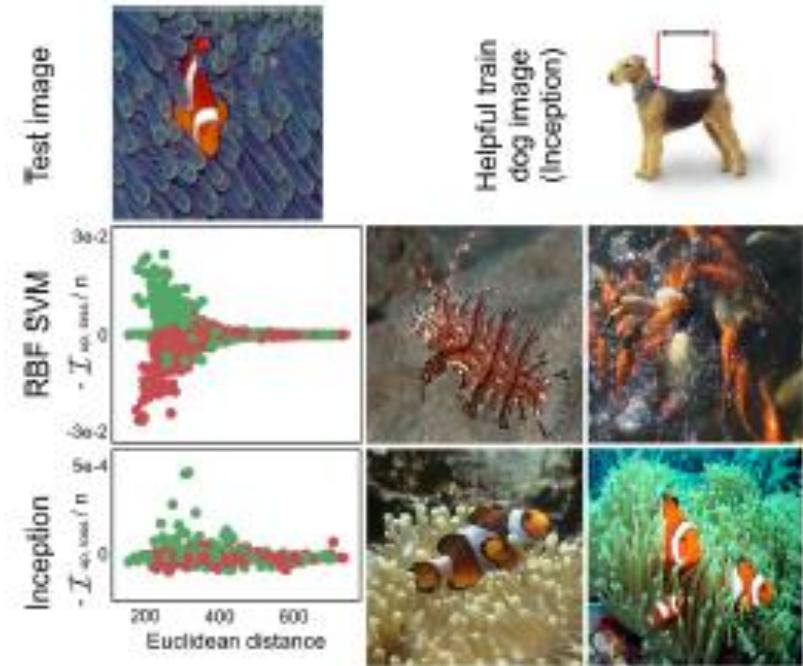
[Koh et. al., Understanding Black-box Predictions
via Influence Functions, ICML'17]



(a) Husky classified as wolf



(b) Explanation



何を考えますか？

- これらすべて「バグ」か？潰さなければならぬのか？
- そもそも潰せるものなのか？それとも技術的限界？
- どうやってこれらに気づくのか？他に対処すべき「同種」や「類似」の状況は列挙できるのか？どうやって？
- そもそも何を持って「品質保証」「完成」とするのか？「仕様」は何なのか？事前に見積・合意できるのか？
- 修正内容をどう決めてどれだけの頻度で更新するのか？
- 顧客やユーザには何をどうやって説明するのか？
- こんな挙動の原因を把握したり、予測し踏まえてテストしたりできるのか？どうやって？開発者ならわかるものなのか？外部品質保証者は何ができるのか？
- . . .

実世界のオープン性による
難しさがある場合も多い

機械学習における本質的な違い

参考：演繹と帰納

■ 演繹

■ 諸前提から論理の規則にしたがって必然的に結論を導き出すこと。普通、一般的原理から特殊な原理や事実を導くことをいう。

■ 帰納

■ 個々の特殊な事実や命題の集まりからそこに共通する性質や関係を取り出し、一般的な命題や法則を導き出すこと。

[大辞林（三省堂） via <https://kotobank.jp/>]

演繹的システム開発と帰納的システム開発

■ 演繹的システム開発（従来）

- 計算や判断を行うための知識・規則（モデル・アルゴリズム）を、人が決めてプログラムという形で書き下す

■ 帰納的システム開発（というか機械学習）

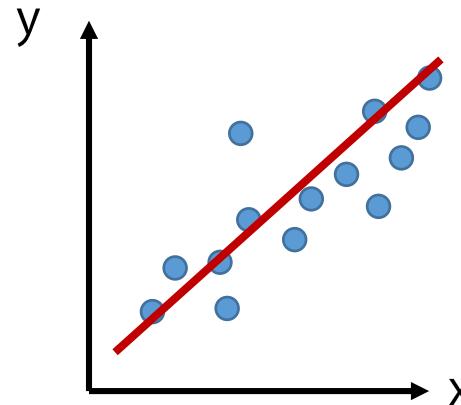
- 計算や判断を行うための知識・規則（モデル・アルゴリズム）を訓練データから獲得し生成する
- それを行うプログラムを人が書き下す

※ 広く「AI」というとどちらの作り方もありうるが、
後者の場合、開発・運用・品質保証が大きく変わる

機械学習：ごくごく簡単に・・・

■ある会社での年齢 x のときの給与 y を予測したい

- $y = ax + b$ と表現できるとして、過去のデータと「一番合う」ように a と b を決めれば、判定・予測プログラムが作れる！



■ 実際は1次関数（パラメータは2つ）では無理

- ディープラーニング（深層学習）では、何十万・何百万ものパラメータを使う

例

この線引きを訓練データから作る

テナガザル



| | | |
|-----------|-----------|-----------|
| 35 24 210 | 20 121 24 | 122 81 20 |
| 211 54 42 | 12 222 90 | 88 79 116 |
| 24 36 98 | 98 181 31 | 66 31 198 |

パンダ



| | | |
|-----------|-----------|-----------|
| 254 32 67 | 222 88 1 | 108 76 14 |
| 12 86 222 | 98 75 122 | 111 74 74 |
| 198 87 33 | 188 173 4 | 68 176 83 |



| | | |
|-----------|-----------|-----------|
| 13 83 33 | 13 45 94 | 75 74 111 |
| 111 873 | 192 1 221 | 237 31 1 |
| 74 35 122 | 93 76 244 | 73 211 45 |



| | | |
|-----------|-----------|-----------|
| 0 245 210 | 20 12 114 | 84 99 100 |
| 11 86 99 | 121 88 91 | 18 0 77 |
| 46 87 121 | 70 76 122 | 122 14 94 |



| | | |
|-----------|-----------|-----------|
| 0 24 31 | 20 21 124 | 12 101 50 |
| 21 54 242 | 112 22 90 | 8 79 214 |
| 124 56 85 | 98 99 141 | 166 1 198 |

機械学習（帰納的システム開発）

■ すごいこと

- 人が明確に規則として書き出せないことも、訓練データが十分にあれば判断・予測などの計算ができる
 - 「この画像はパンダの画像だ」
 - 「本Aを買った人は、本Bも買う可能性が高い」
- 訓練データを更新していくけば、新しいことに対応できる
 - 今月デビューした芸能人の画像も判別可能に
 - ユーザごとのクセがある音声指示も認識可能に

機械学習（帰納的システム開発）

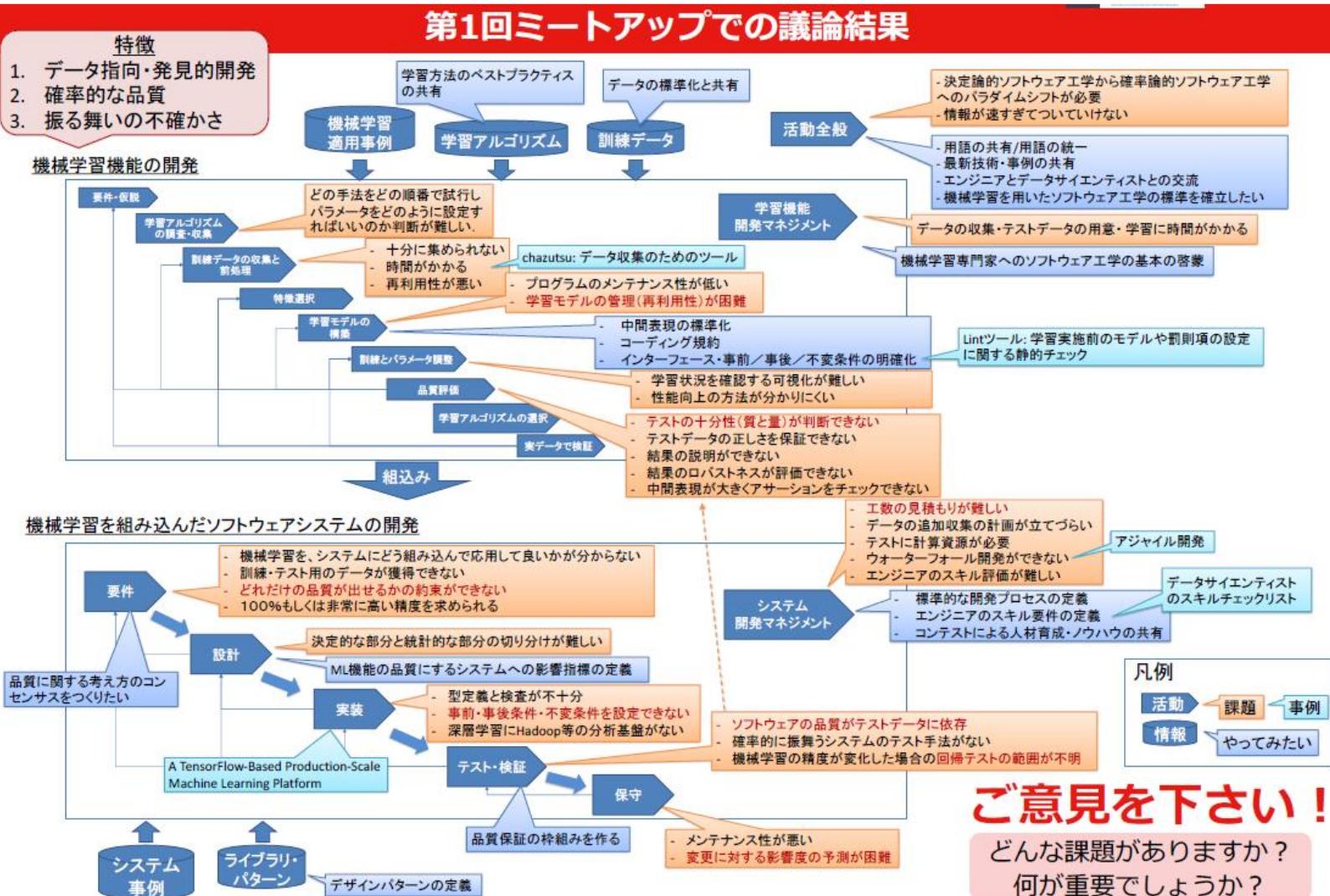
■ 大変なこと（主に）

- 原則として機能は不完全（100%正解は出せない）
- どの程度の性能が出るか作ってみるまでわからない

- 大量かつ「適切な」訓練データが必要
- 訓練データがあればうまくいくとは限らない
- 訓練データ外のデータ、テストしていないデータで
どう振る舞うかは未知（かなり似たデータでさえ）

- ある出力がなぜ起きたのかは説明できないことが多い
(その情報がない、あっても複雑で把握できない、人が
言葉にできない傾向をとらえていることも)

2017年秋のワークショップより



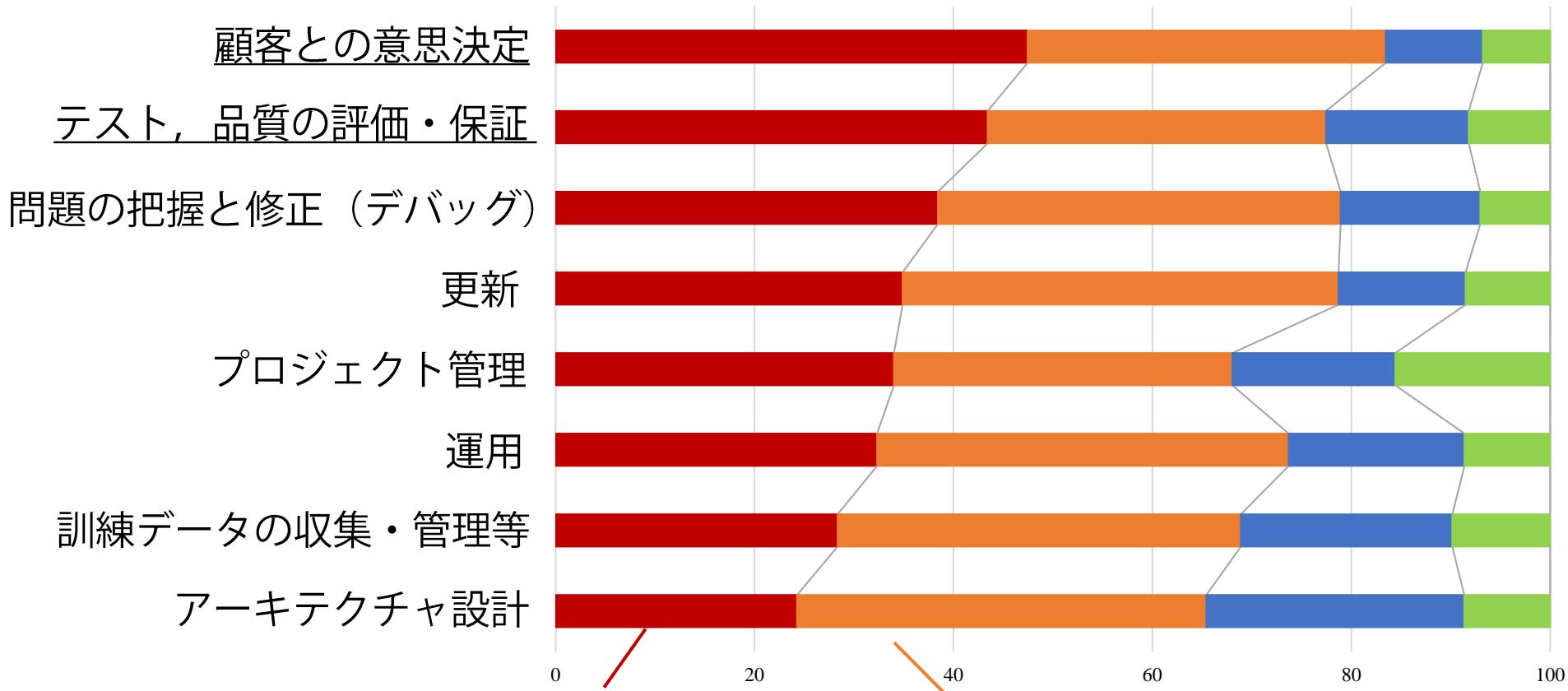
ご意見下さい！

どんな課題がありますか？
何が重要でしょうか？

[吉岡ら, 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, 2017]
[\[http://research.nii.ac.jp/~f-ishikawa/work/fose17/ \]](http://research.nii.ac.jp/~f-ishikawa/work/fose17/)

2018年のアンケートより (MLSE)

- 280名弱のアンケート対象者、大半は開発者
(ソフトウェア開発経験豊富、機械学習には新規参入)



根本的に異なる新たな考え方が必要

[<https://sites.google.com/view/sig-mlse/>] → 「発行文献」

考え方は同じだが
手法／ツールが未成熟

おまけ：性能（精度）の難しさ

何を「100%」の基準とするか？

「100%」にはならない、どう受け入れるか？

■特にオープンな実世界を入力・動作環境とする場合

■例：自動運転のための画像認識

→「10万件の画像でテストしました！」

「霧の日は試した？」 「山道は？」 「逆光は？」

しかも、人間にとって意味のあるこれらの区分は、

機械学習で作ったモデルには意味がないかもしれない！

（言葉にしがたい「不得意画像」があるかも）



おまけ：
鏡面タンクローリー

[<http://passage.eshizuoka.jp/e831305.html>]

おまけ：
リアケース
[P. Koopman 2018]



おまけ：保守の課題

■ 従来のソフトウェアとはだいぶ性質が異なる 「技術的負債」の存在

[Sculley et al., Machine Learning: The High-Interest Credit Card of Technical Debt, 2014]

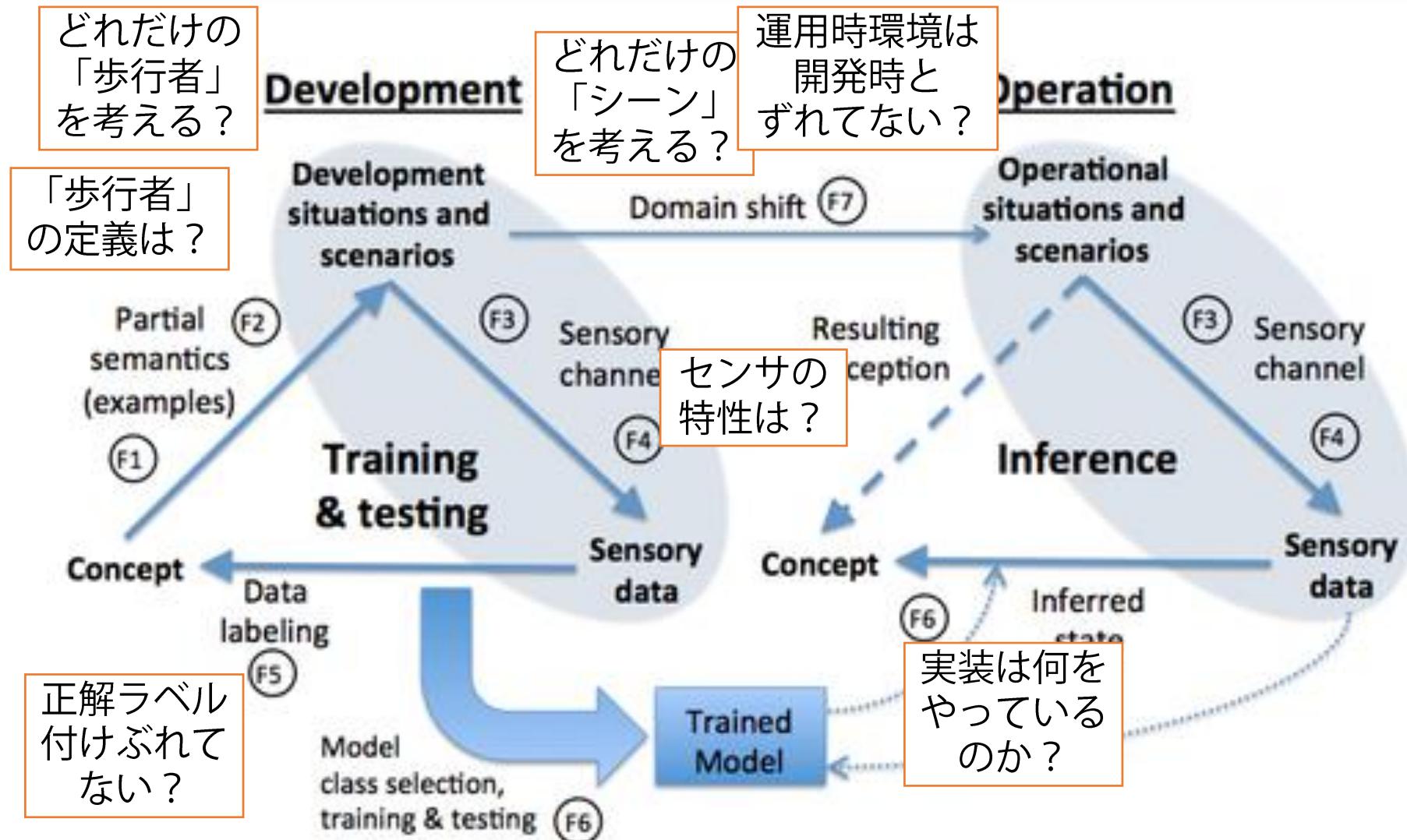
Googleでの経験から14個の負債を紹介
「機械学習は高利息クレジットカード」

「取り急ぎリリースへ（整理などを後回し）」の弊害や
時間経過での劣化が大きく、従来と種類・対応法が異なる

■ 例：一つの入力の傾向が変わると全てが変わる可能性がある（「ここだけ変えれば済む」と言えない）

► 「何とかVer. 1.0リリース」は借金の始まり・・・

おまけ：自動運転の識別器での「不確かさ」



[<https://uwaterloo.ca/waterloo-intelligent-systems-engineering-lab/projects/assuredai-safety-assurance-ai-based-automated-driving>]

機械学習に対する テスト・検証技術の追求

従来のテストアプローチが通じない！（1）

■そもそも「テスト不可能」

- 画像分類など、正解を与えることのコストが大きい
- 給与判断など、唯一の正解を決めがたい場合もある
- 推薦やデータマイニングなどの場合、未知の正解を求めることが目的で、出力の期待値は存在しない

■「テストでバグを見つける」？

- 分類や予測が100%成功することは本来ないため、期待値と実際の出力を比べて Pass/Fail を確認しても、「Failならバグの存在を示唆」ということにはならない（訓練データの問題？訓練コードの問題？性能限界？）

※ 「テスト不可能」という語は1982年にはあった！

[Weyuker, On Testing Non-Testable Programs, 1982]

従来のテストアプローチが通じない！（2）

■ 単体テスト？

- 実装は大きなブラックボックスであるため、「小さい単位でテストしてバグの存在にすぐ気づき、バグを探す範囲も小さくすることができる」ということができない

■ 要求カバレッジ？ 同値クラス？

- 要求・環境がオープンな場合、要求の列挙は困難
- なお、「要求に対し、実装のここが対応している」という関係は把握できない

■ コードカバレッジ？

- 実装コードの分岐カバレッジ等は数件のテストで100%になる（分岐ではなく「数値」が出力を決める）

先端企業から出ている原則・指針の例

「仮定・想定をテスト」せよ」は方向性の一つ

(従来の「テスト」より広義)

■ ベストプラクティス集

- 例：データの統計を追跡するなどして、問題として顕在化しない失敗を見張れ

[Zinkevich, Rules of Machine Learning: Best Practices for ML Engineering, 2016]

■ どれだけ「テスト」できているかの評価スコア

- 例：個々のfeatureの入力値範囲や分布が予想と合うか
- 例：直接的なメトリクス（精度等）と実影響のあるメトリクス（クリック率等）との相関はどうか、例えばあえて前者を悪くして比較（A/Bテスト）するとどうなる？

[Breck, What's your ML Test Score? A rubric for ML production systems, 2016]

最近の論文例 (SE系・不完全・Core Aのみ)

■ 形式検証や安全性保証

- SMTソルバを用いた形式検証 [CAV'17]
- 抽象解釈を用いた検証 [ICML'17, NIPS'18]
- 安全な強化学習 [AAAI'18]

■ 「問題」があるケースの探索を行うテスティング

- ニューロンカバレッジを用いたサーチベースドテスティング [SOSP'17] [ICSE'18]
- 確率的ゲームによる検証 [TACAS'18]
- より密なカバレッジとそれを用いたサーチベースドテスティング [ASE'18]
- 公平性のテスティング [ASE'18]
- 「驚き」に基づくテスティング [ICSE'19]
- ミューターション（人工バグ）を用いた敵対的サンプル検出 [ICSE'19]

■ 「バグ」（コーディングミスなど）の発見・修正

- メタモルフィックテスティングの実証 [ISSTA'18]
- バグの種類に関する調査 [ISSTA'18]
- 過学習・未学習に対する修正（デバッグ手法） [FSE'18]
- 深層学習ライブラリにおけるバグ位置推定 [ICSE'19]

■ テストの評価・修正（再訓練）

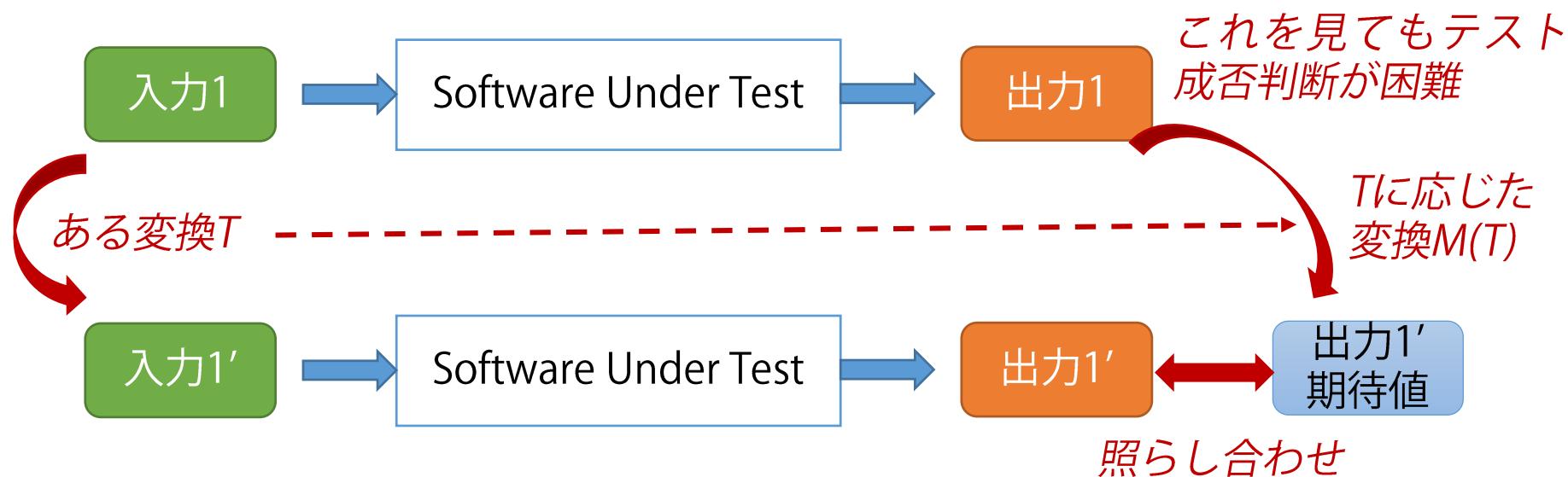
- ミューターション分析（テストの強さの評価） Mutation analysis [ISSRE'18]

背景技術：メタモルフィックテスティング

■ メタモルフィックテスティング

- 多数の入力を用いてテストをしたくても、各出力の正しさ（テスト成否）を判定するのが困難または高コスト

→ 「入力を変えると出力はこう変わるはず」という関係を検証、既存テストケースから多数のテストケースを生成



例： $\sin(x) = \sin(\pi - x)$

[Segura et al., A Survey on Metamorphic Testing, 2016]

メタモルフィックテスティングの適用

■ 訓練アルゴリズム, 訓練済みモデル, システム全体のテストへの適用事例

(ある入力で出力を出してみた後に)

- ランキング生成：入力購買データから1位商品を抜く
- 時系列分析：入力信号をすべて定数値ずらす
- ドローン探索プランニング：地図を回転させる
- 画像処理：画像のRGBを入れ替える
- . . .

[Murphy et al., Improving the Dependability of Machine Learning Applications, 2008]

[Jarman et al., Metamorphic Testing for Adobe Data Analytics Software, 2017]

[Lindvall et al., Metamorphic Model-based Testing of Autonomous Systems, 2017]

[中島, データセット多様性のソフトウェア・テスティング, 2017]

[Dwarakanath, Identifying Implementation Bugs in Machine Learning, 2018]

サーチベースドテスティングの適用

- 機械学習モデル（画像識別器や進路判断器等）のサーチベースドテスティング
 - 既存画像に雨や覆い等を加えることで様々な入力を作る
 - 「ニューロンカバレッジ」を最大化するテストスイートを出す、つまり「ホワイトボックス・テスティング」
 - これによりテストスイートの多様性・網羅性を上げつつ、「悪いケース」を探す
 - 例：同じ入力を別バージョンの実装に入れたときと比べて、より大きく出力結果が変わるケース



1.1 original



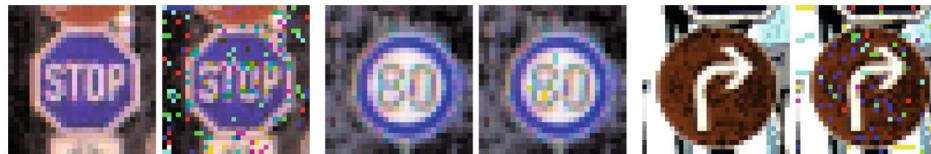
1.2 with added rain

[Pei et al., DeepXplore: Automated Whitebox Testing of Deep Learning Systems, 2017]

画像元： [Tian et al., DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars, 2018]

形式検証技術の適用

- 形式検証技術による頑健性検証
 - 画像認識において、「一定範囲の画像操作」を行っても認識結果が変わらないかどうかを網羅的に検証
 - SMTソルバーや抽象解釈を応用

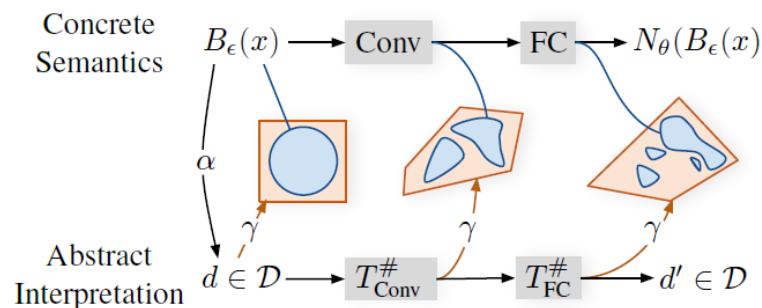


“stop”
to “30m speed limit”

“80m speed limit”
to “30m speed limit”

“go right”
to “go straight”

画像元：[Huang et al., Safety Verification of Deep Neural Networks, 2017]

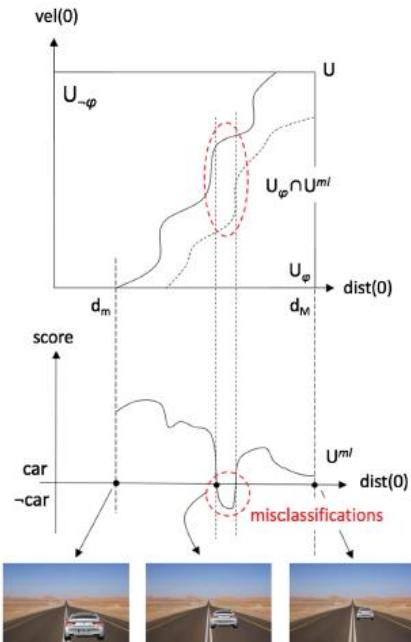


画像元：[Mirman et al., Differentiable Abstract Interpretation for Provably Robust Neural Networks, 2018]

システムレベルの要求に基づくテストの例

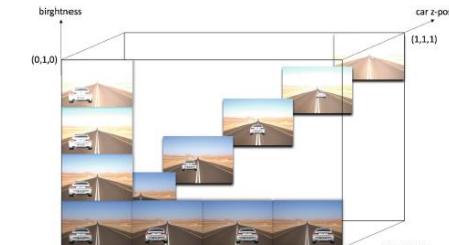
■ システム全体の要求を踏まえるのが重要

- 機械学習部品（生成されたモデル）の信頼性だけむやみに突き詰めてもしようがない（どうせ不完全だし）
- 例：遠くの物体を誤認識しても衝突には至らないかも



例題：自動ブレーキシステム（人の操作や先行車の位置が入力、「先行車との距離が一定以上ある」ことが要求）

1. 完璧な機械学習部品を用いると要求を満たすが、常に失敗する機械学習部品を使うとそうならないような入力パラメータ領域を絞り込む
2. その領域に限定して機械学習部品が誤認識するような画像を探す
3. その画像を用い要求を満たさないケースを探す



[Dreossi et al., Compositional Falsification of Cyber-Physical Systems with Machine Learning Components, 2017]

補足：QA4AIガイドライン（1）

■ 5つの評価軸（簡単なチェックリスト）

- データ, モデル, システム
- プロセス
- これらが顧客の期待と合致

■ 技術力タログ (Body Of Knowledge)

- 機械学習の標準的な性能評価指標（精度等）
- 敵対的サンプル・その探索テスティング
- メタモルフィックテスティング
- ニューロンカバレッジ
- 説明可能性

※ 現状のベスト・今後も継続的に更新

補足：QA4AIガイドライン（2）

■各ドメインでの踏み込んだ分析

- 自動運転：不確かさの基でのリスク軽減プロセス
- 産業プロセス：ステークホルダーの多様性、環境への強い依存性、顧客の納得の必要性
- スマートスピーカー：異なる抽象度の要求、自然言語入力の多様性
- 生成系システム：自然さなど、AIで実現されるような高度な品質評価機能

まとめ

- 産業界の高いニーズ
- 研究者にとっても学術的な追求
 - 一時的流行・バズワード（呼び変え）でもない
- まだBlue Ocean・手探り
 - MNIST, CIFAR10くらいでの評価が多い
 - 今年のICSEで、ニューロンカバレッジに対する疑義を投げかける論文が出るなど、「何をすべきか？何に実効的な意味があるのか？」という問い合わせのレベル、合意・確立はこれから

[Li et al., Structural Coverage Criteria for Neural Networks Could Be Misleading, 2019]