CRDS-FY2017-WR-11

俯瞰ワークショップ報告書

機械学習型システム開発へのパラダイム転換

平成29年12月8日（金）開催

国立研究開発法人科学技術振興機構 研究開発戦略センター
Center for Research and Development Strategy, Japan Science and Technology Agency

エグゼクティブサマリー

国立研究開発法人 科学技術振興機構（JST）研究開発戦略センター（CRDS）は、国の科学技術イノベーション政策に関する調査・分析・提案を中立的な立場に立って行う組織として、今後わが国が取り組むべき重要な研究領域に関する提言を行ってきている。また、その提言に向けた前段階として、広く技術分野の状況を俯瞰し、注目すべき研究動向の抽出を行ってきている。これら俯瞰や提言は、文部科学省・内閣府をはじめとする政府の各種政策立案のためのインプットとなる。

今回、システム・情報科学技術分野の俯瞰の一環として「機械学習型システム開発へのパラダイム転換」に注目した。そして、2017 年 12 月 8 日（金）に、この研究動向の現状・課題や今後の取り組みの方向性を把握するためにワークショップを開催した。

以下では、「機械学習型システム開発へのパラダイム転換」に関する問題意識、および、ワークショップでの議論の要点を報告する。

現在、人工知能の第 3 次ブームと言われるが、これを牽引しているのは、深層学習（ディープラーニング：Deep Learning）をはじめとする機械学習技術である。そして、単に言葉が流行っているのではなく、機械学習技術は様々なシステムに埋め込まれ、実社会での応用・実用化が急速に広がっている。

この機械学習技術はシステムの動作を大量データから帰納的に定義するものであり、プログラムコードを書いてシステムの動作を定義する従来の開発スタイルとは大きく異なっている。そのため、従来ソフトウェア工学として構築・整備されてきた方法論・技術は必ずしも使えず、システムの要件定義や動作保証・品質保証にも新しい考え方が必要になる。システム開発のパラダイム転換が起きているのである。

このパラダイム転換によって、システム開発に必要な人材スキルや方法論が刷新され、この変化に追従できないと、ソフトウェア産業やシステムインタグレーターは競争力を失いかねない。また、動作保証・品質保証等の考え方が整備されないまま、機械学習技術を組み込んだシステムの社会実装が急激に進むと、そこで発生した問題や事故が思わぬ社会問題化する懸念もある。

このような産業界にも大きなインパクトを与え得る「機械学習型システム開発へのパラダイム転換」に際し、機械学習技術そのものの研究開発が活発なのに比べて、それを組み込んだシステム開発の技術・方法論の研究開発については、まだ十分な取り組みが推進されていないように思える。

本ワークショップでは、上述の問題意識を共有し、次の 3 点を目的として議論した。

① 機械学習型システム開発へのパラダイム転換における課題の全体像をつかむ
② それらの課題に対する取り組みがどこまで進んでいるか、状況を把握する
③ 取り組みを加速･強化するための方針･施策についてアイデアを集める

まず CRDS からワークショップの趣旨説明を行い、問題意識の共有と事前調査に基づく課題の全体像のイメージを示した。次に、外部から招いた 5 名（丸山宏氏、中島震氏、谷

幹也氏、藤吉弘亘氏、佐藤洋行氏）に発表いただき、この問題に対する考え方や具体的な取り組み事例・状況等を把握した。それを踏まえて、上記①②③に関する総合討論を行った。総合討論の中では、③に関連して、JST の未来社会創造事業の枠組みも紹介し、その活用の可能性についても意見を交換した。

比較的早い段階から今回の問題を認識し、取り組みを進めている丸山宏氏（株式会社 Preferred Networks 最高戦略責任者、日本ソフトウェア科学会理事長）と中島震氏（国立情報学研究所 教授）の二人からは、それぞれの立場から機械学習型システム開発とその課題の全体観が示された。丸山氏は主に機械学習技術の視点から、中島氏はソフトウェア工学の視点から、機械学習型システム開発において、従来のソフトウェア工学的なアプローチと異なる考え方が必要になる理由（本質的に確率的であること、解消できない不確かさを持つこと）も論じた。

続いて、谷幹也氏（日本電気株式会社 セキュリティ研究所長）、藤吉弘亘氏（中部大学 教授）、佐藤洋行氏（株式会社ブレインパッド マーケティングプラットフォーム本部 副本部長）からは、実際の応用やビジネスにおいて直面する問題や対応策について、具体的な事例を含めて紹介された。谷氏は従来型のシステム開発によるビジネスから機械学習型に移行する立場であり、佐藤氏は当初から機械学習型でビジネスに取り組んできた立場である。藤吉氏は機械学習型システム開発の典型的な応用分野の一つであるロボット制御の事例を取り上げた。

5 人からの発表と総合討論を通して、前述の 3 点について以下のような知見を得た。

①の課題の全体像は、システム開発のフローの視点とシステムの構成要素の視点から整理できる。機械学習は本質的に確率的で、解消できない不確かさを持つため、機械学習を用いて作られたシステムの動作は、原理的に「100%の保証」はできない。そのため、従来のソフトウェア工学的なアプローチは使えず、システムの品質管理に新しい考え方・アプローチが必要であることをはじめ、多くのチャレンジングな技術課題がある。

②の取り組み状況については、システム開発契約や製造物責任法（PL 法）に関わる懸念も含め、産業界での問題意識が非常に高まっている。しかし、システム開発の現場においてはまだ解決策は見えておらず、これを解決する技術開発に対する強いニーズがある。このような市場や産業界の問題意識の高まりに呼応するようにワークショップが繰り返し企画される等、ソフトウェア工学系の研究コミュニティにおける取り組みが進み始めた。一方、AI 研究コミュニティの側では、機械学習型システム開発の方法論そのものへの取り組みがまだ手薄だが、機械学習型システム開発に関わる課題を別な側面からとらえたような重要な研究課題（深層学習の脆弱性・ブラックボックス問題、帰納型と演繹型の統合等）への取り組みが進んでいる。

③の取り組みの進め方や方針・施策については、ソフトウェア工学分野と AI 分野のクロスした新分野で、研究開発への取り組みと研究者育成が必要である。また、技術課題以外に、制度・ガイドライン面の整備（機械学習型システムの品質基準作り、機械学習技術に関わる知財権、プライバシーや倫理の問題等）も重要である。

目　次

1. 趣旨説明

ワークショップ開催趣旨

福島 俊一（科学技術振興機構 研究開発戦略センター）

ワークショップのテーマとして掲げた「機械学習型システム開発へのパラダイム転換」に関して、まず私から問題意識や本日議論したい点が何かを説明する。

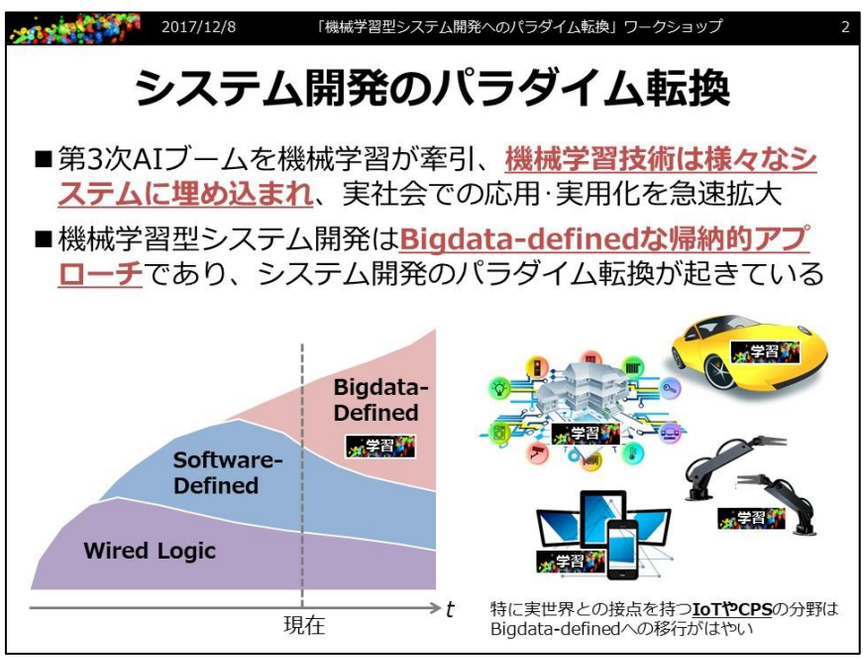


図 1.1　システム開発のパラダイム転換

システム開発のパラダイム転換がまた起きつつある（図 1.1）。人工知能（AI）の第 3 次ブームと言われるが、機械学習技術、特に深層学習技術がそれを牽引している。スマホ、産業用ロボット、車の自動運転、スマートホームやオフィス環境等、いろいろなところにどんどん機械学習が組み込まれていくが、この機械学習を組み込んだシステムの作り方は従来と異なるものになる。

過去から見ていくと、まず Wired Logic、回路の形でシステムの動作・ロジックが実現されていた。次に Software-defined、プログラムのコードを書くという形でシステムの動作が定義された。そして、新たに Bigdata-defined、これが機械学習を用いたシステムの作り方で、データによってシステムの動作を定義することになる。パラダイム転換と言っても、以前のものがなくなって、後のものにすべてが置き換わるという形ではなく、一つのシステムがこれらを組み合わせて作られるようになっていき、3 種類の比率が急速に変わってきているという形である。

今回 Bigdata-defined と呼んだ機械学習型システム開発は、データによってシステムの動作を定義するタイプで、例えば、センサデータからいま実世界がどんな状況にあるかを判断して、こういう状況だったらこんなことをする、というような実世界の状況に応じて動作するシステムとも相性が良い。つまり、IoT や CPS（Cyber Physical System）等、

最近市場が盛り上がっている実世界接点の分野は、Bigdata-defined への移行が進みやすい。

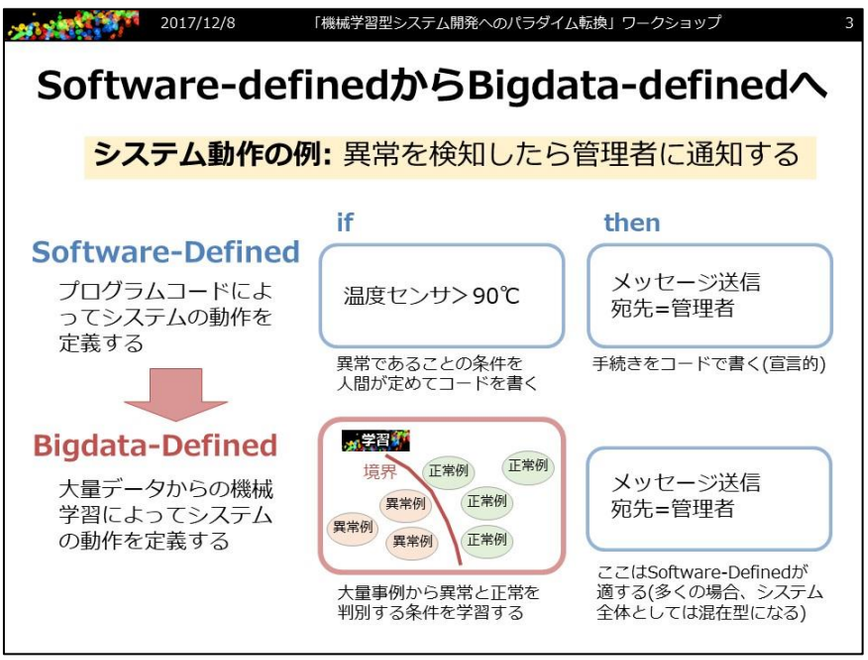


図 1.2　Software-defined から Bigdata-defined へ

従来の Software-defined と機械学習による Bigdata-defined では、システムの作り方がどう変わるかを、ごく簡単な例で説明する（図 1.2）。例えば「異常を検知したら管理者に通知する」というシステムを考えると、Software-defined ではプログラムのコードを書く。「if ～ then ～」の形で、「if ～」の方に明確に定義した「異常な状況」の条件、例えば「温度センサーが 90 度を超えたとき」というような条件を書く。そして、「then ～」の方に、メッセージを誰々に送るといった手続きを書くという、というのが一例である。

一方、Bigdata-defined では、異常の条件をデータで定義する。正常の例、異常の例を多数のサンプルとして与えて、その境界をデータから学習する。そして、その学習された境界と比較することで、新たに入ってきたデータが異常・正常のどちらのケースかを判別し、異常のケースに該当したら、メッセージを送る手続きを起動する。このメッセージを送る手続きはコードを書くタイプでよく、この部分は Software-defined になる。つまり、システムの中で Bigdata-defined と Software-defined は混在する形になる。

では、このパラダイム転換に対応できないと、どのようなネガティブインパクト、社会的・産業的ダメージを被るか（図 1.3）。

まず、Bigdata-defined の世界では、従来の Software-defined の世界で確立されてきた従来のシステム開発方法論が使えない。従来の開発方法論では、仕様が最初に明確にあって、それをトップダウンに分割して設計していく、論理的なモデルをプログラムコードとして実装していく。それでできたプログラムに対して、分割された単体のテストをして、それを組み上げて結合テストをしていくというような方法論ができあがっている。しかし、Bigdata-defined な機械学習型システム開発の場合は、そもそも要求仕様が明確に書けない

とか、分割してそれを組み上げるような考えが成り立たないとかで、従来の方法論が適用できない。それによって、システムを開発する人材に要求されるスキルや方法論も刷新される。これに追従できないと、産業界が競争力を失い、非常に大きな打撃を受けることになる。というわけで、とても重要な課題だと認識している。

図 1.3　対応遅れが招く社会的・産業的ダメージ

もう一つ、社会的な面でも問題がある。品質保証・動作保証の問題。最近、深層学習の脆弱性の問題が話題になっている。車の自動運転の際に、標識の認識に深層学習を使うことが考えられるが、「止まれ」（STOP）の標識を認識できる深層学習モジュールに対して、「止まれ」標識に少しのノイズを乗せたものを見せて、速度制限の標識だと騙すことができたという報告がある。事前の学習データとして、あらゆる場合を尽くすことは無理なので、こういった攻撃ができてしまう。動作や品質が保証できないならば、製造物責任（PL 法）やリコール制度に該当しないか等も絡んできて、大きな問題になる。

このような問題意識を共有した上で、本日のワークショップで議論したいポイントはこの 3 点である。

1. 機械学習型システム開発へのパラダイム転換における課題の全体像をつかむ
2. それらの課題に対する取り組みがどこまで進んでいるか、状況を把握する
3. 取り組みを加速･強化するための方針･施策についてのアイデアを集める

1 点目は、いろんな問題が指摘され始めているが、それを整理・共有したい。実際の研究の中で、あるいは、事業の現場で、具体的な課題が見えてきていると思うので、それら

を共有し、Bigdata-defined という新しいシステム開発のパラダイムに移るにあたって、どこにどんな問題が出てくるかの全体像をつかみたい。

2 点目は、そういった課題に対して、現場で少しずつ取り組みが始まっていると思うので、その取り組みの事例をご紹介いただく。そこから、課題に対してどれくらいまで取り組みが進んでいて、どんなところに大きな課題が残っているか、状況を把握したい。

このような課題に対する取り組みをさらに加速・強化したいと思っているわけだが、そのための具体的な施策や進め方の方針についてアドバイスをいただきたいというのが 3 点目。その乗り物として、JST の未来社会創造事業が候補になりそうだと考えているので、今回、その事業の枠組みも紹介しつつ、ご意見・アイデアを集めたい。

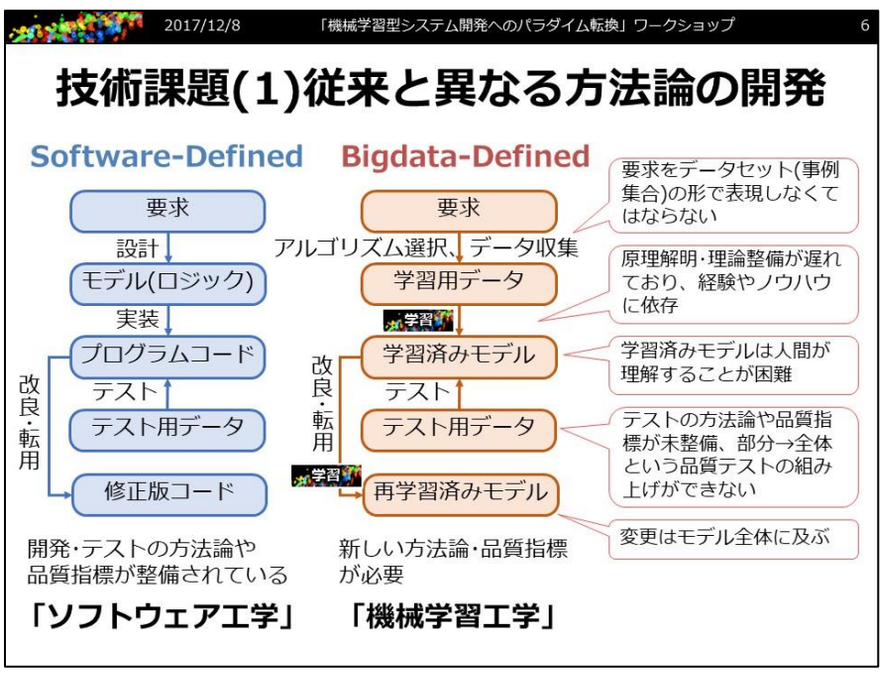


図 1.4　従来課題(1)従来と異なる方法論の開発

1 点目としてあげた課題の全体像については、私の方で仮の絵を描いてみた（図 1.4 と図 1.5）。今回の問題意識に関して、既に十数名の有識者からヒアリングや議論をさせていただいており、その結果に基づいて私なりの整理をしたものである。あくまでたたき台なので、本日の発表や議論を通して改良していきたい。

二つの絵を描いたが、一つめは従来の方法論との対比という視点で描いたもの（図 1.4）。プログラムコードを書いてシステムの動作を定義する従来の Software-defined の方法論が左側、それと対比する形で、機械学習を用いてデータによってシステムの動作を定義する Bigdata-defined のケースを描いた。そして、各ステップについて、こんな課題がありそうだというものを、右の方に吹き出しの形で置いた。

従来の Software-defined を振り返ると、まず要求仕様があって、それをモデルというか、ロジックとしてどう実現していくかを考えて、具体的なプログラムコードとして実装する。そして、できたプログラムコードに対して、要求仕様やモデルに基づいたテスト用データを用意して、テストをして品質を高める。これは新規に作る場合の基本的な流れで、実際

には既に作られているプログラムを改良したり、流用したりというケースが多々ある。プログラムの場合、既に作られたものがモジュール性高くできていれば、システムの改良の際には、別のモジュールを追加したり、あるモジュールのみ修正したりと、局所的な修正によって改良版ができる。以上のような開発・テストの流れや方法論、品質管理の考え方や品質の指標というものが整備されていて、学問的にもソフトウェア工学といった学問分野として確立されてきている。

これに対して、Bigdata-defined、機械学習型システム開発に同じような流れを当てはめようとすると、だいぶ違ってきて、従来の方法論が使えない。具体的には、まず、要求からシステムを作るという考え方に関しては、要求を仕様の形で書き出すのではなく、学習用データセットで仕様が決まる。つまり、要求をデータセットの形に表現して与えなければいけない。さらに、与えたデータセットから深層学習によってどんなモデルができるかに関しても、基本的な原理解明や理論整備はまだ十分できていない。ある種、勘と経験とノウハウによっている。また、でき上がった学習済みモデルは、深層学習等では人間が理解できるものではない。この解釈性が欠けているという問題は品質管理の上で問題になってくる。テストについても、学習用データとして与えたものと、テスト用データとして与えるものをどういう関係で考えたら、適切なテストができるのかとか。あるいは、テストは、従来は単体テストから結合テストへと組み上げていく方法が有効だったが、機械学習型ではそれが使えない。流用についても、学習済みモデルを流用することは可能だが、追加で学習させていったときに、局所的な修整にならずにモデルの全体に変更が及んでくるといった特性がある。

このように、Bigdata-defined な機械学習型システム開発には、従来の考え方が使えず、新しい方法論や品質指標が必要になる。従来の考え方は「ソフトウェア工学」として確立・整備されてきたが、これに対して、機械学習型システム開発では「機械学習工学」を確立・整備していこうと、丸山さんが提唱している。

もう一つ、構成要素の視点から絵を描いてみた（図 1.5）。これは中島さんが描かれた絵をベースに、私の方で追記したもの。

ここでは、構成要素をざっくり次のようにとらえた。まず、機械学習方式というアルゴリズムがあって、それがプログラムとして実装される。そのプログラムの中身には学習処理と推論処理がある。学習処理は、データセットからモデルを作るステップで、推論処理は、学習済みモデルを使って、新たに入ってくるデータを処理するステップである。また、実際のシステムは、機械学習プログラムだけでなく、従来型のソフトウェアと組み合わせる形で全体が構成される。

これらとデータセットを対応付けると、学習ステップに対しては、初期の学習のためにデータセットを与えて学習するというのがまずあり、さらに、環境変化等によって追加で学習させる必要が生じたら追加学習用のデータセットを用意することになる。それから、テストのためのデータセットがある。一つは学習処理自身が正しく動くかというテストであり、もう一つはでき上がったモデルに対するテストである。

このような構成に基づいて、機械学習型システム開発の課題をあげる。まず、アルゴリズムのレベルでは、精度を確保するために効率を犠牲にするとか、精度と解釈性がトレー

ドオフだとか、このバランスをどう考えるかが、基本的なところで、システムの頑健性・安全性・信頼性に関わってくる。次に、機械学習プログラムとして実装する段階では、そもそも、もとのアルゴリズムをきちんと実装できているのかとか、ある種、正しさの基準や、その検証をどうするかといった点に課題がある。

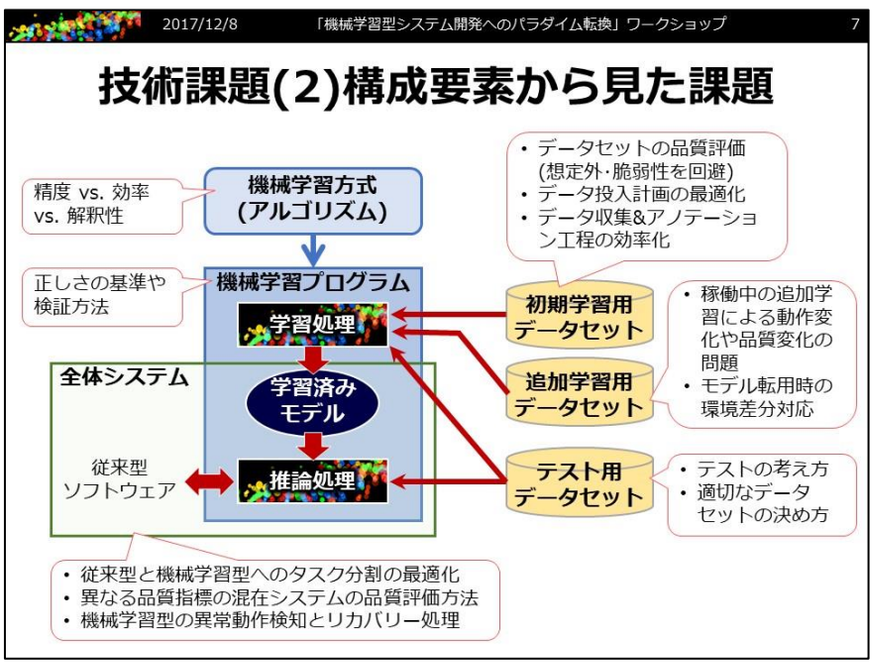


図 1.5　従来課題⑵構成要素から見た課題

それから、システム全体として組み上がった段階では、またいろいろな観点からいろいろな課題が出てくる。従来型と機械学習型のものが混在するので、どう分担するか、つまり、全体システムの目的に対してタスクをどう分割し、どういうタスクは従来型を使い、どういうタスクは機械学習型を使うのが最適な構成になるか。分担できたとしても、従来型のソフトウェアには品質管理をする手法がある一方、機械学習型の方にはない、あるいは、従来型とはまったく別の品質管理の考え方をするとなったとき、それらが組み合わさったシステム全体の品質をどう評価し、どう管理するのか。さらに、機械学習型では、どうしても 100%の動作保証は難しいので、何らかの問題が起きたときのことも考える必要がある。そういう異常状況を検知してリカバリーするとか、いろいろ考えなければならない。

機械学習型システム開発における学習用データやテスト用データの準備や与え方についても、いろいろ考えねばならない。データセットはなるべく想定外が起こらないようなカバレッジの高いデータセットを与えたいが、そのためのデータセット自体の品質評価方法を考える必要がある。データを投入する順番によって効率も変わってくるかもしれないので、その最適化はどうするのがよいか。また、データも実際はただのセンシングデータではなくて、正解データやアノテーション情報が付いている必要があるわけで、そのようなデータを効率よく作成する方法、これはツール開発の話かもしれない。

さらには、追加学習に関する問題。機械学習の使い方として最初に作った学習済みモデルを使い続けるという運用形態もあるかもしれないが、実際の環境では状況が変化していくので、学習済みモデルが合わなくなり、精度・性能が落ちていく。そこで、稼働しなが

ら新しいデータで追加学習して、どんどんシステムが変わっていくというような運用形態もあり得る。そのとき、動作が変わっていってしまうシステムの品質をどう考えるか。また、学習済みモデルを転用したとき、学習時の環境と実際の稼働環境が異なる場合、その差分をどう考えていくか等、さまざまな課題がある。

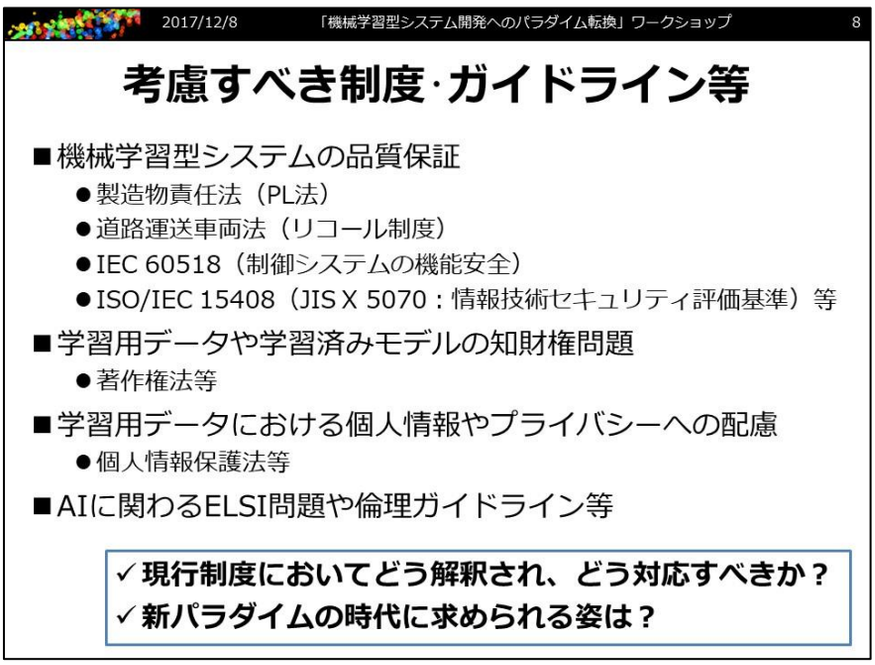


図 1.6　考慮すべき制度・ガイドライン等

また、技術課題とは別に、法律・制度、ガイドラインといった面も検討していく必要がある（図 1.6）。今回は項目だけあげておくが、品質保証については、製造物責任法（PL法）やリコール制度、品質基準の標準との関係等を考えておく必要がある。また、知財権の問題、個人情報やプライバシー、倫理問題等も絡んでくる。

ただし、既に存在するいろいろな現行ルールにおいて、どう解釈され、どう対処するかという面だけでなく、新しいパラダイムへの転換であることを踏まえて、そういう時代に合わせて、どういう姿が求められるかといった観点で、新しい枠組みを考えていくことも必要だと思っている。

以上で私からの趣旨説明を終わる。

この後、5 人の先生方から発表していただく。前半の 2 人、丸山さんと中島さんは全体観から話していただけると思う。一方、後半の 3 人、谷さんと藤吉さんと佐藤さんは、実際の応用やビジネスに取り組まれている中で直面している問題や解決へのアプローチについて、具体的な話が聞けると思う。

その後、全体討議に入るが、その冒頭では JST 未来社会創造事業の紹介も行う。その上で、先ほど私の発表の中で示した三つのポイントを中心に議論を進めたいと思う。

2. 発表

2.1 機械学習工学に向けて

丸山 宏（株式会社 Preferred Networks）

機械学習型システム開発はシステム開発の新しい方法の一つと思っている。機械学習というが、基本は深層学習（ディープラーニング）がユニバーサルな機械学習のため、深層学習を考えていただきたい。深層学習が何かと非常に簡単にいえば、ある入力 X を送って出力 Y を出す関数、Y = f(X)である。ただし、恐らく入力が 1 万とか 10 万とか極めて高い次元のベクトルである。出力は判別問題であれば低次元だが、生成問題であれば高次元のものが出てくる。これが API、例えば Google の API などで使う限りはこのぐらいの理解でよいが、問題はこの中身をどう作るかというところ。

今までのプログラミングで関数を作る場合は、要件定義をして、その要件を先験的な知識に基づいてモデル化する。例えば摂氏から華氏へ変換するという場合には、摂氏から華氏への変換をモデル化し、モデル化できれば、それを段階的に詳細化して実装していく。これが普通の関数の作り方である（図 2.1.1）。

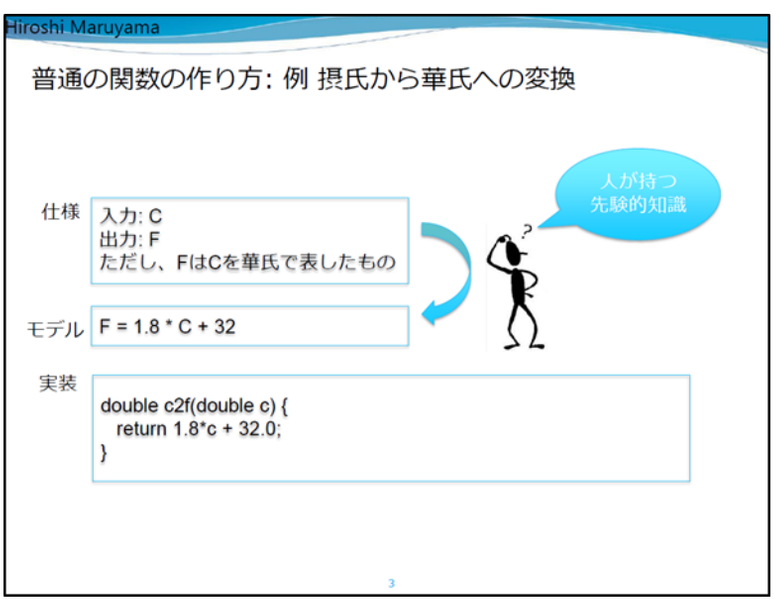


図 2.1.1　普通の関数の作り方

一方、私たちが考えている深層学習ではどう関数を作るかというと、例えば摂氏から華氏の変換の場合、摂氏と華氏の温度計を買ってきて、時々、温度を観測する。そして、観測結果から図のようなサンプルの訓練データセットを作る。この訓練データセットからは、あとはほぼ自動的に訓練が行われてある関数ができる。この入出力をまねするものができるというのが深層学習によるプログラミングのスタイルである（図 2.1.2）。

深層学習はユニバーサルな機械学習であると言ったのは、深層学習は非常にパラメーター数が多いために、任意の非線形の関数を任意の精度で近似することができる。もう少しはっきり言えば、任意の関数に対してある深層ニューラルネットがあり、その深層ニュー

ラルネットがその関数を近似できるという意味である。そういう意味では、ユニバーサルなプログラミングのパラダイムということが言える。

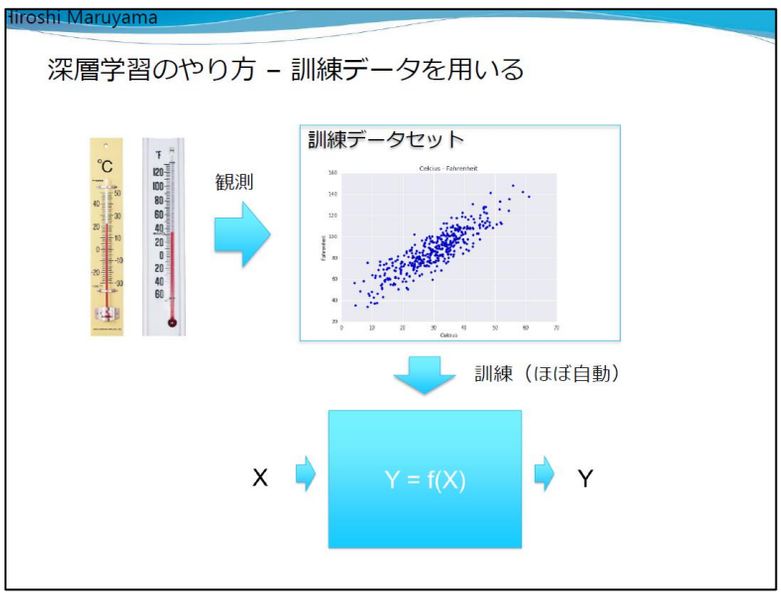


図 2.1.2　深層学習による関数の作り方

問題は、深層学習を含む統計的機械学習には本質的には限界がいくつかあることであり、今回は３つだけお話しする。

第一は、統計的機械学習は皆そうだが、過去のデータに基づいて未来を予測する。そのため、過去と未来が同じ確率分布に沿っているときしかうまくいかない。杉山将先生（理化学研究所革新知能統合研究センター長）は、少しこの条件が緩くなるような研究をされているが、基本的にはこうである。

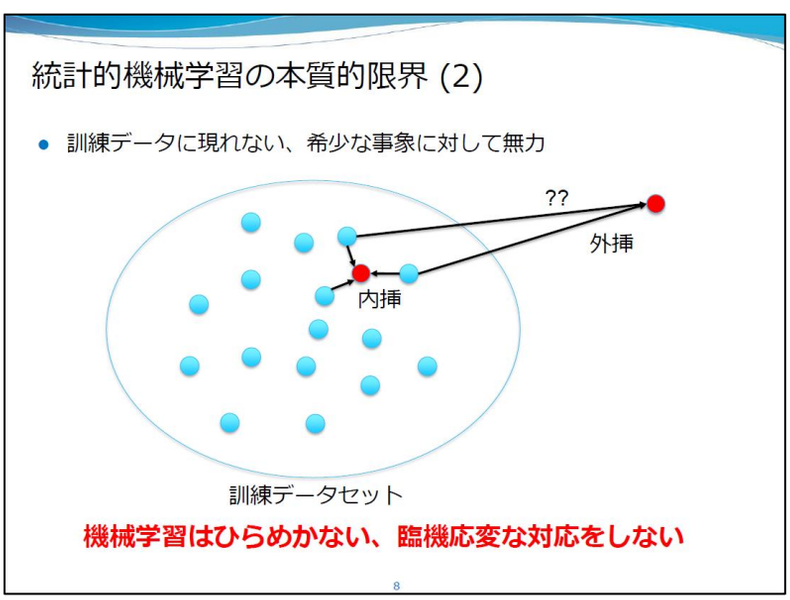


図 2.1.3　統計的機械学習の本質的限界

第二は、たとえ過去と未来の確率分布が同じであったとしても、訓練データセットにあらわれない領域のデータについては正しく予測できない。訓練データセットの近くにある点に関しては、いわゆる内挿は、非常にうまく近似できる。訓練データセットに出てこないような領域のデータに関してはほとんど無力である。そのため、時々、深層学習で非常事態に対応するようなシステムを作ってくださいという方がいるが、それは無理である。人工知能という言葉を使うと何か人のひらめきとか、臨機応変ということを考えがちだが、そういうことはできない（図 2.1.3）。

第三は、恐らく一番これが重要なところだと思うが、そもそも統計的機械学習は本質的に確率的である。私たちは常に何かの確率分布を仮定するが、そこからランダムサンプリングして訓練データセットが出たと仮定する。ここから先は、深層学習のアルゴリズムでモデルができる。問題は最初のステップのランダムサンプリングのところにある。ここがランダムサンプリングであるため、統計的機械学習は確率的となる。

例えばサイコロを 100 回振って、100 回とも 1 が出るということはほとんどないはずだが、絶対にないとは言い切れない。そういう意味で、必ず訓練データセットにはバイアスが出てくる。このため、深層学習に基づくシステムは 100%正しいといったことは原理的に言えないというところが非常に重要なポイントである。

以上が深層学習を含む統計的機械学習の限界だが、続いて一般的に深層学習（機械学習）を使ったシステム開発がどのように行われているかお話しする（図 2.1.4）。

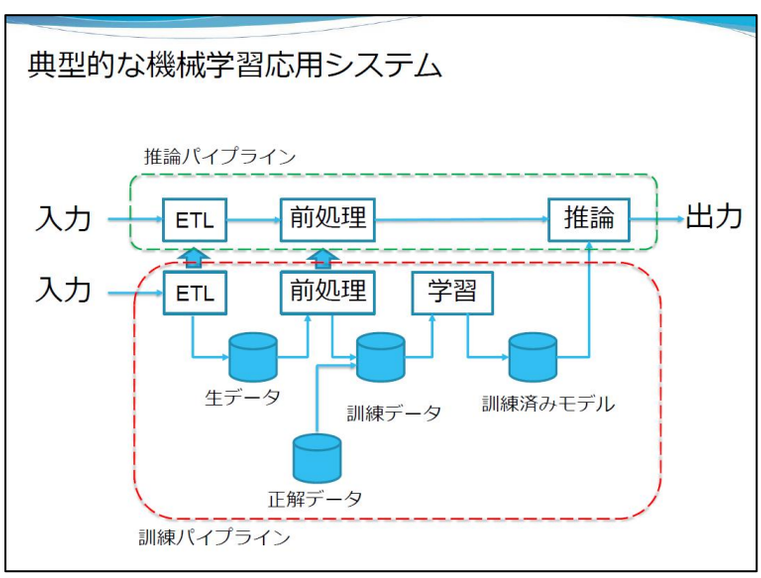


図 2.1.4　典型的な機械学習システム

大事なことはまず二つのパイプラインがあるということ。下側が訓練パイプライン、上側が推論パイプラインと呼ばれる。入力があり、それを ETL（Extract、Transform、Load）することで生データができる。これに対して、例えば欠測値を埋めるとか、外れ値を除外するといった前処理を加える。さらに多くの場合には正解ラベルをつけるような人手の作業が発生する。こうした前処理により、訓練データセットができ、それを学習アルゴリズムにかけて学習することで、訓練済みデータセットができる。それを使って推論のパイプラインが動く。ここで気をつけなくてはいけないのは、このモデルは、ETL、前処理に依

存しているため、全く同じ ETL と前処理をしなくてはいけない。これは必ずしも、いつでもできるものではなくて、それが精度に影響する。

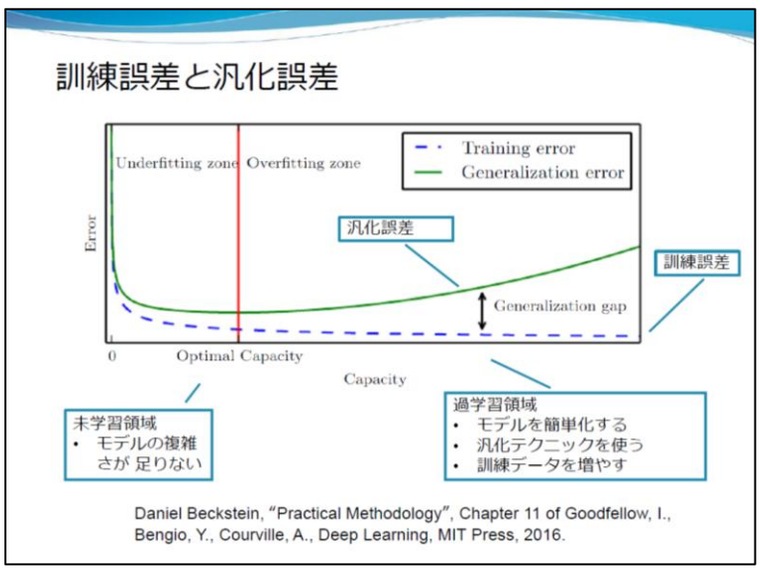


図 2.1.5　訓練誤差と汎化誤差

学習のところだけを見ると、二つの誤差を気にする必要がある（図 2.1.5）。一つは訓練誤差である。訓練データセットをいかに再現できるかというのが点線のグラフ。これは誤差なので、右にいけばいくほど小さくなり、訓練データセットをよりよく表現できるようになる。しかし、訓練データセットは有限のため全部を覚えておけば、ここは 100%再現可能なものができるはずだが、それを行うと今まで見ていなかったデータに対する精度が悪くなる。

今まで見ていなかったデータというのは汎化誤差と呼ばれるもの。モデルが複雑になればなるほど訓練誤差は小さくなっていくが、汎化誤差は大きくなっていく。汎化誤差が大きくなると、例えばモデルを簡単化するとか、汎化テクニックを入れるとか、訓練データセットを大きくするということをする。ここのうまいポイントをいろいろ調整しながら決めるのは、今のところ、エンジニアリングではなくてアートの世界であり、非常に難しいと言われている。

ビジネスの面から見てもっと難しいのは、こういうシステムの開発は今までのシステムの開発と違い、非常に探索的であること。そもそも、お客様の期待値をどうコントロールするかということが難しい。お客様は大抵、プルーフ・オブ・コンセプトをやっても、その先はなかなかいかないというようなことがある。

こういうことを考えると、こんなことをできる人はどのぐらいいるのかという話になり、典型的な反応は AI 人材が足りないということを言われる。そのため、人材育成に力を入れようということが言われる。しかし、ちょっと待てよと思うかもしれないが、1960 年代にソフトウェア危機ということが言われたことがあった。IBM 社の System360 という汎用機ができたとき、こんなに高性能なコンピュータのソフトウェアを書く人がいない、ソフトウェアクライシスと言われた。

当時のプログラミングというのは、アセンブリーのプログラミングでありそんなことができる人はほとんどいなかった。それがいわゆるソフトウェア工学の始まりだった。そこから、FORTRAN やオブジェクト指向が出てきた。それと、同じことが私たちにも必要ではないかというのが私の強い問題意識である。

最近、幸いなことに福島さんを初めいろいろな方の意識が高まってきた。11 月には機械学習×ソフトウェア工学というミートアップを行った。その結果を 12 月ソフトウェア工学ワークショップで発表した。来年 1 月に情報処理学会のソフトウェア工学研究会のウィンターワークショップで、その専用のセッションを開くこととなった。

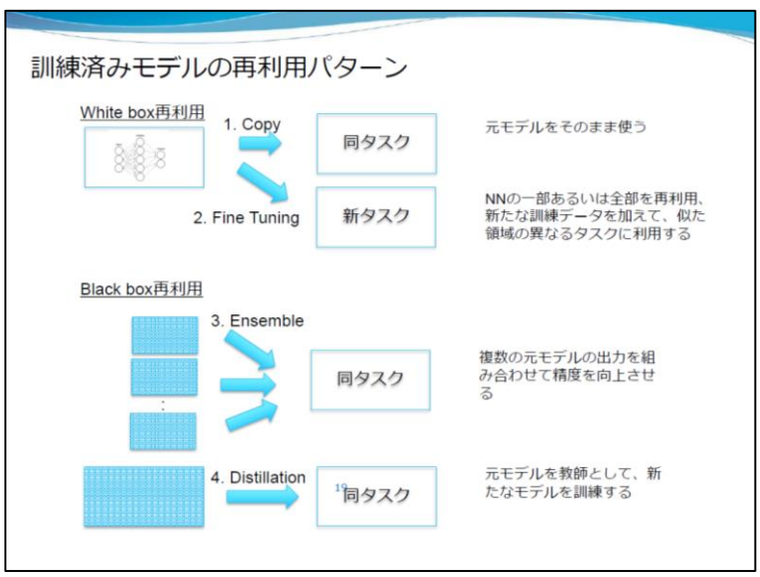


図 2.1.6　訓練済みモデルの再利用パターン

この機械学習工学で何を議論したらいいかということについて、少なくとも私の見るところで三つ大事なテーマがある。

第一のテーマは、成果物の再利用である（図 2.1.6）。ソフトウェア工学の歴史は再利用の歴史であり、機械学習でも再利用を考えないといけない。特に大事なのは訓練済みモデルだが、訓練済みモデルをそのままコピーして使うのはいわゆるホワイトボックス再利用で、それは普通のソフトウェアの再利用と同じである（図 2.1.6 の 1 番目のパターン）。違うのは、プラスアルファのデータを加えることで、少し違うドメインにチューニングすることができること。ファインチューニング（Fine Tuning）とか、トランスファーラーニングと呼ばれるが、これが 2 番目のパターンの再利用である。

実はホワイトボックスだけでなく、ブラックボックスの再利用もうまくできるパターンがある。一つはアンサンブル（Ensemble）と呼ばれているもので 3 番目のパターンである。同じタスクに対する複数のモデルが少しずつ性能の違うものがあった場合に、それの平均をとると実は個別のモデルよりも性能が理論的に高くなることが知られている。

ブラックボックス再利用にはもっと変ったものがあり、ディスティレーション（Distillation）と呼ばれる 4 番目のパターンである。あるモデルがあったときに、それに対してその API がわかっていたとする。そうするとその API をたたく。私のデータセットで繰り返したたくと、それの正解ラベルが出てくる。そこで、出てきたものを新たな訓練

データセットとして、私の別のモデルを訓練するということをする。そうすると、出てきたものの性能は似たような性能のものが出てくるが、中身は全く違うものができる。これをコピーと呼ぶか、呼ばないかについては全くコンセンサスがない。これをどう守るかというルールの問題もあり、こういうことをどうやっていくかという問題もある。そのため、機械学習は普通のプログラムとは違った再利用のパターンがあるということである。

この訓練済みモデルを再利用するための標準化の動きが少しずつ進んでいる。一つはNNEF、（Neural Network Exchange Format）という KHRONOS GROUP が制定しているもので、恐らく今月（2017 年 12 月）にドラフトが出てくるはずである。もう一つは、マイクロソフトとフェースブックがやっている ONNX というもの。こちらは、グラフを Protobuf（Protocol Buffers）の形で書き下したというだけの非常にある意味、クイック・アンド・ダーティなやり方。こちらのほうが実は実装が早くて、私たちの Chainer も間もなくこれをサポートするはずである。ここまでいけば、トレーニングに使ったフレームワークと、それから、インフルエンスに使ったフレームワークを別のものにするということが簡単にできるようになる。

そういうことを実は８月に、メルボルンの IJCAI（International Joint Conference on Artificial Intelligence）でワークショップを開催し議論した。そこでの議論で一番おもしろかったのは、日本の著作権 47 条 7 項[1]について。これは著作物の統計情報をとることについては、もとの著作権を侵さないということが明示的に書かれている。そのため、例えばスターウォーズの DVD をたくさん買ってきて、そこから深層学習でスターウォーズライクな映画を生成するというモデルを作っても、日本でやる限りは著作権的に問題にならない。

第二のテーマは、いつも聞かれるが、品質保証である。これについては先ほど言ったように、100%という話がもともとない仕組みのため、この話をすると特に組込み系の方々は、深層学習は自分の話ではないとなる。難しいのは特にこれである。Google の人たちが書いた機械学習システムは高金利クレジットカードだという論文[2]がある。この中に、CACE（Changing Anything Changes Everything）という原理が書かれている。先ほどのパイプラインがあるが、それぞれのソフトウェアやデータは、それぞれ、１個ずつは個別にはテストできない。しかし、そういうものが必ず最終的な入力、出力の精度に影響を与えているため、どれか１個を変えても全体の精度は変わるので、非常に難しいということ。

それから、テストするときに先ほど訓練データの話があったがもう少し細かく見ると訓練用のデータセットと、バリデーションセットという汎化性能をテストするためのデータセットにまず分ける必要がある。これで訓練して、汎化性能がどのぐらいかをチェックする。それを繰り返し眺めながら、少しずつパラメーターのチューニングをしていくという

[1] 著作権法 47 条 7 項　「著作物は、電子計算機による情報解析（多数の著作物その他の大量の情報から、当該情報を構成する言語、音、影像その他の要素に係る情報を抽出し、比較、分類その他の統計的な解析を行うことをいう。以下この条において同じ。）を行うことを目的とする場合には、必要と認められる限度において、記録媒体への記録又は翻案（これにより創作した二次的著作物の記録を含む。）を行うことができる。ただし、情報解析を行う者の用に供するために作成されたデータベースの著作物については、この限りでない。」

[2] Machine Learning: The High-Interest Credit Card of Technical Debt
https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43146.pdf

ことが深層学習のやり方だが、これをやっている限り、このデータは両方とも学習に使われている。学習に使われているということはどういうことかというと、でき上がったモデルは全体のデータに対して最適化された、ある意味、過学習されたモデルになる。そのため、それを世の中に持っていって、全く見たことのないデータに突き当てると、十分な性能が出ないということが起きる。

そこで、何をすべきかというと、そもそも、深層学習をやる前に、データセットの中から学習に使われないデータを取り分けておき、これを評価用に使うべきである。これをバリデーションセットと呼ぶのは非常に間違いで、ソフトウェア工学的には、これはベリフィケーションと呼び、評価用データの方をバリデーションと呼ぶべきである。いずれにせよ、この評価用のデータに関しては、学習をしている人は見てはいけない。つまり、ここに情報の漏れがあると、正しく評価できないということが知られている。ここは、実は深層学習の研究者も結構、いいかげんなことをやっていることがあるが、ここは真剣に考える必要がある（図 2.1.7）。

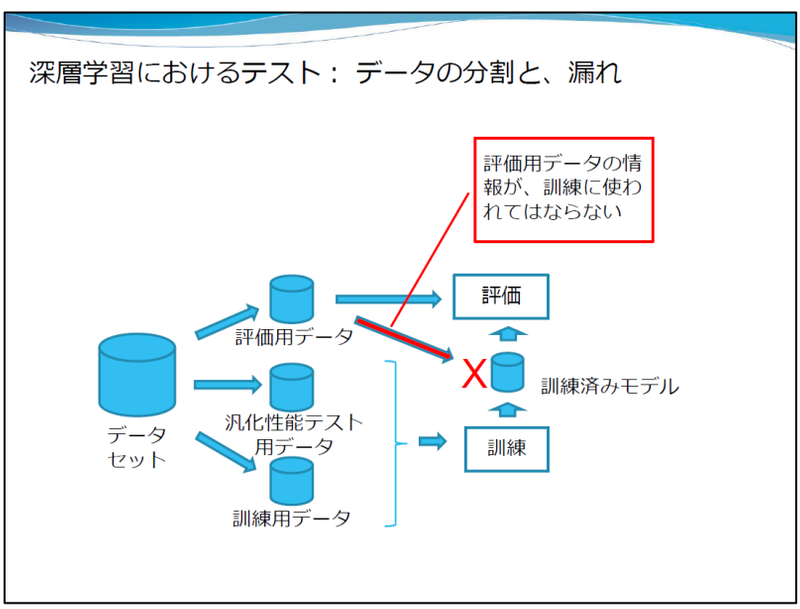


図 2.1.7　深層学習におけるテスト

深層学習のシステムが事故を起こすと、「ほら見ろ」という話になる。では、V字型開発をすれば 100%正しいのか、安全なのかということを問いたいと思う。なぜならば、ソフトウェア工学をやっている人はよく知っているが、ソフトウェアには必ずバグがある。大体 1,000 行に 1 行くらいのアベレージであるということが知られている。

softrel.com のウェブサイト[3]に典型的なバグ密度の数字が掲載されている。このサイトに Average fielded defect density という、ソフトウェアが製品としてフィールドに出てから見つかったバグの密度が載っているが、1,000 行に対して 0.4 程の数字である。セーフティクリティカルなシステムに対しても似たような数字である。実はV字型開発をしている人たちは、実際にはパフォーマンスとして 100%正しくないものを出している。そういう意

[3] http://www.softrel.com/Current%20defect%20density%20statistics.pdf

味では、深層学習と大して変わらないだろうと思う。

さらに主張したいのは、V字型開発によるプログラムの品質保証はどうしているかというとプロセス品質である。プロセス品質とは、このプログラムはこういうプロセスで作ったので品質は高いはずという言い方である。プロセス品質指標の一つは、どのくらい開発中にレビューをしたか。製品の品質をレビューしているのではない。単にそのプログラムの開発プロセスがどうだったかということをレビューしているだけである。そういう非常に間接的な指標が今、世の中で使われているソフトウェア品質の指標である。

それに対して先ほど申し上げたように、評価用のデータセットを第三者機関が絶対、開発者に漏れないように評価する仕組みがもしできれば、これは客観的にプロダクトの品質を評価できるはずである。そういう意味では、品質評価に対して機械学習に基づくシステムは、今までのやり方よりももっと直接的にプロダクトの品質をはかれるはずだと思う。

第三のテーマは、要求の話である。要求がどのように効いてくるかというと、例えば、私どもは強化学習でぶつからない車というデモを作っている。ぶつかったときに与えるペナルティを無限大にすると、動かない車ができる。普通、システムを作るときに、車が動くということは暗黙に開発者が理解しているが、機械学習は、そこを明示的に定量化しないといけなくなる。

同じような議論がIJCAIで行われており、ロボットにコーヒーをとってきてというと、下の階のスターバックスに行き、客がたくさん並んでいるので全員撃ち殺してコーヒーを持って帰ってきた。何で人を殺したか聞くと、ロボットはそう言われなかったからと答えた。そういう話だが、機械学習を使う限りは、これは最適化の問題なので、最適化の効用関数を要求としてきちんと定義しなくてはいけない。それを人が全部きちんとやるのはものすごく難しい。そういう意味で、正しい仕様のあり方というのを提起したいと思う。

あと、1点だけお話しするが、深層学習でやると、必ずそれは説明不能だと文句を言われる。どのノードが何に対応するかという後づけの理由を言ったりするが、曖昧な言い方になる。ただし、ここでもう一つ私が議論したいのは、深層学習もデジタルコンピュータで動いているプログラムであり、同じ入力、同じプログラムを与えれば必ず同じ結果が出る。ただ、全ての何百万もあるパラメーター一つずつ全部が、微妙にアウトプットに効いているので、それを説明したとしても普通の人には理解できないというだけで、説明は完璧に可能である。

しかし、普通の人が「説明」と言うのは、納得性のある説明のことを言っているはずである。納得性のある説明とは何かというと、まだきちんと定義されていない状態である。もし納得性というものを数値化して定量化して定義できれば、納得性に対して最適化する深層学習を作り、納得性のある説明を出すシステムというのは、作れるのではないかとも思う。

いずれにせよ、今までのシステムの作り方と深層学習による作り方というのは、どちらか一択という話ではなくて必ず組み合わせになる。ただし、モデルがよくわかっているときには、今までの演繹的なシステムの作り方をしたらいいと思う。

モデルが複雑過ぎて書き下せない場合には、深層学習でやってはどうか。これもノウハウとして機械学習工学の中に蓄えられていくような話だと思っている（図 2.1.8）。

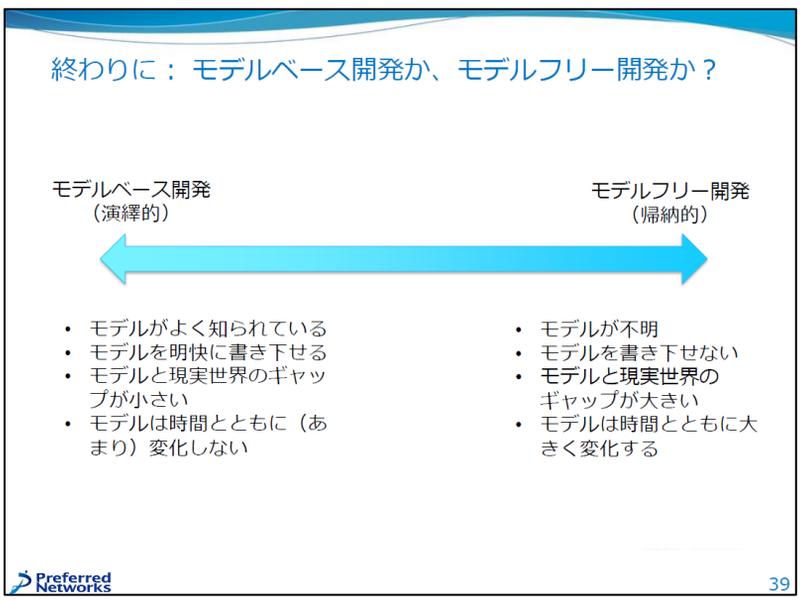


図 2.1.8　モデルベース開発とモデルフリー開発

【質疑応答】

Q：工学というと大体それに対応する科学がある。例えば物性物理があれば電子工学がある。機械学習工学に対応するサイエンスの部分というのは統計学か。
A：ベースは統計学のように思う。

Q：機械学習工学について、特に 3 点を中心に問題の切り口を言っていただいたが、それに対する実際の世の中の取り組みはどういう状況か。
A：一番進んでいるのは Google だと思う。Google の人たちは、ソフトウェアエンジニアリングのマインドがすごくあって、機械学習をやっているので、論文もそういうものがたくさん出ている。

Q：人間の知能みたいなことは、深層学習だけで実現できそうな気はしないと思うが、それは正しいか。
A：そのとおりだと思う。私たちはここで人工知能の話をするべきではなくて、あくまでも深層学習あるいは統計的機械学習の話をすべき。
Q：統計的というのはビッグデータ的なものをイメージするが、それは正しいか。
A：ビッグデータの定義にもよるが、そう思っていい。

Q：個別の指示や個別の例を指示して、よりよくなるというシステムもあり得ると思うがどうか。深層学習的でも統計的な学習でもないように思うが。
A：帰納的ではあって統計的ではないような学習の仕組みというのは考えられる。

Q：AI 人材が不足しているという話の際に、人材が要るのか要らないのかという言及はなかったが、機械学習工学をやるという人材自体は、要るということでよいか。
A：そのとおり。ただ、やみくもに深層学習の Chainer の使い方を教えるのではなく、方法論がまずあって、それを教えていかないといけない。
Q：そうすると、先ほど話が出たような相当する科学をやる人材を育て、そこの工学にいくというようなステップを踏まないといけないのではないか。
A：そこはアカデミアの側で頑張っていただかないと。

Q：統計学を知っていれば機械学習ができるかは、違うように思う。統計的機械学習は、統計的な手法を使うが、統計学そのものではない。その辺が機械学習工学として新たに何かベースとしてアカデミックとか、大学風にいうとカリキュラムとして何か出てくるのではないか。
A：コンピュータサイエンスと統計学の融合領域みたいなのが必要かもしれない。

2．2 ソフトウェア工学×機械学習
～ディペンダブルな機械学習ソフトウェア・システム開発に向けて～

中島 震（国立情報学研究所）

専門はソフトウェア工学である。タイトルが「ソフトウェア工学×機械学習」となっているのは、新しいパラダイムにしても、いきなり新しいものが突然ゼロから現れるわけではない。そこで本日は、ものごとの流れを見ながら、どう捉えていくか、どう捉えるべきか、ということを考えようと思う。

自分自身は機械学習を専門としていない。２年ほど前にふと、専門家は深層学習のプログラムのデバッグをどうやっているのだろうと思い何人かの人に聞いたところ、結局、デバッグらしいこと、ソフトウェア工学的にはプログラムの正しさを確認するという作業を、全くやっていないということを知り驚いた。機械学習的なものはかなり前から始まっていたにもかかわらずやっていないのである。

新しいソフトウェアの流れは CPS（Cyber-Physical Systems）であると考えている。2006年に米国で始まった。欧州では agenda CPS というドイツの議論を経て、今は Horizon 2020で取り上げられている。Smart World Vision を実現するテクノロジーという見方で、FP6ならびに FP7 の成果と新しいビジョンの間をつなぐものだ。Smart World Vision として、例えばドイツでは「考える工場」というのをやっている。IoT＋AI とあるが IoT＋ML の方がいい。また、マイケル・ポーターは IoT という言葉でさまざまな提言をしている。中でも重要なことは、今がソフトウェアとデータの時代であるということである（図 2.2.1）。

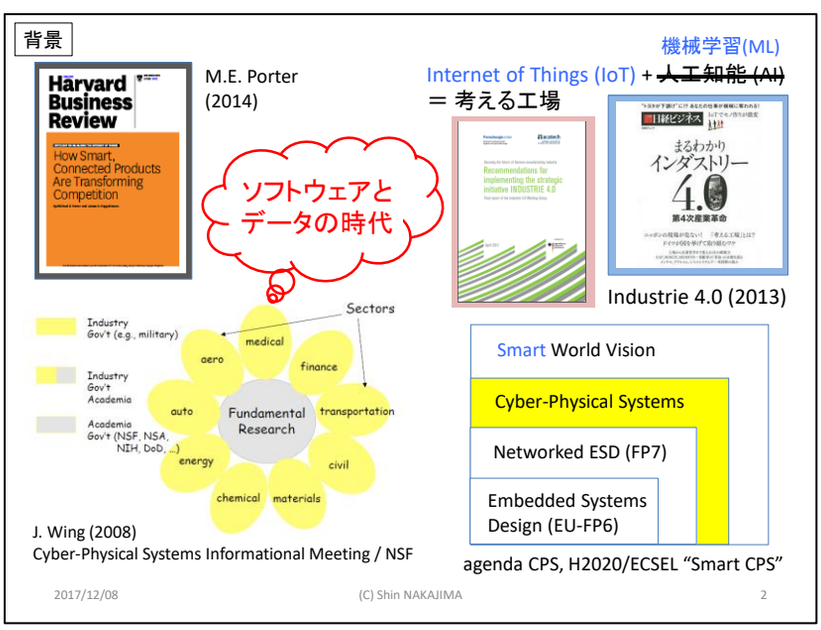


図 2.2.1　背景

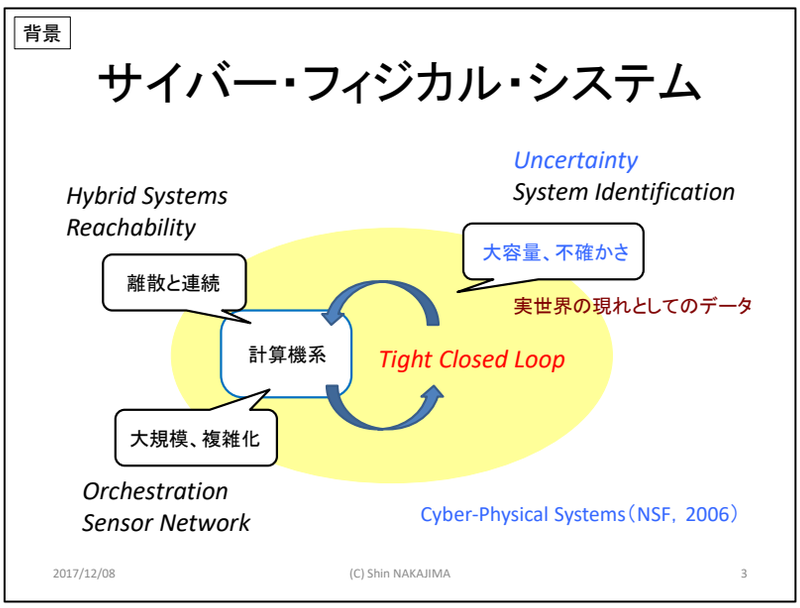


図 2.2.2　［背景］サイバー・フィジカル・システム

CPS についてはいろいろな理解をする方がいるが、僕は、計算機系と外の世界があり、そこに Tight Closed Loop があると考えている。CPS の分野には 3 つの基本的な研究対象があると言われている（図 2.2.2)。一つ目は離散と連続である。ハイブリッドシステムのようなものである。計算機システムはデジタルなので、アナログの情報も含めハイブリッドで取り扱わなければいけない。

CPS の前に SEC（Software Enabled Control）があり、制御工学をコンピュータライズするということだった。そのときの一つのポイントがオーケストレーションである。これが二つ目である。１個の制御系で対象をコントロールするのではなく、それらがさまざまにつながり大規模、複雑化したものを対象にする。その技術トレンドは CPS につながる。

三つ目は大容量と不確かさである。計算機の外側の世界とつなごうとするとき、SEC の時代は、制御工学の考え方なので、システム同定が問題だった。扱うデータは不確実さを持つので、大容量と同時に不確かなデータを扱わなければいけない。このようなデータを扱わなければいけないのは、制御工学にしても、CPS にしても組込みシステムにしても必然である。なぜなら、実世界の現象というのはデータとしてしかシステムに取り込まれないから。そのデータが離散的なもの、１個１個のものか、あるいは連続的なものか、あるいは大容量なものか、それによってテクノロジーが変わる。したがって、ここで大容量・不確かになった場合について考えると、今、流行している機械学習は、CPS を支える一つの要素技術だとみることができる。

ソフトウェアの品質保証について述べる（図 2.2.3)。なお、スライドの左上の小さな字、ここでは SE は、ソフトウェアエンジニアリングの見方であることを示している。ソフトウェアエンジニアリングは図 2.2.3 の四角で囲った部分だけであり、実際は非常に狭い技術である。

まず、何を作るかということが出発点である。その後、どうやって作るかという品質の作り込みの工程を経てプログラムを作る。次に、作ったプログラムが作ろうと思っていた

ものに合っているかどうかを調べる品質の確認の工程にいく。これがＶ字型と呼ばれる工程の説明である。ポイントは、確認はプログラムを対象に確認するということである。このとき正しさは設計仕様書からくる。プログラムを対象にすること、正解値がわかっていること、この２点がキーポイントである。

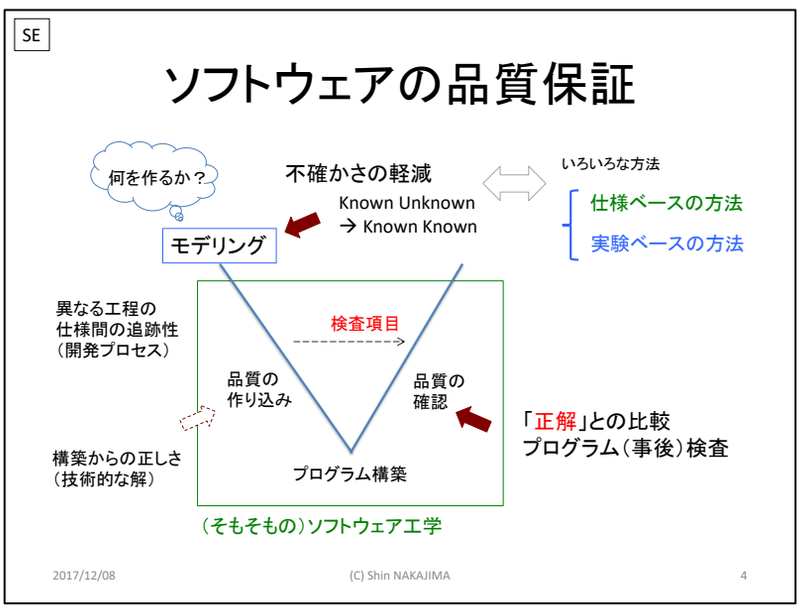


図 2.2.3　［SE］ソフトウェアの品質保証

問題は、何を作るかという出発点のモデリングは、実はソフトウェア工学の外側の技術であるということである。古いソフトウェア工学の教科書には、システムエンジニアリングがやるべきことであると書いてある。その後、紆余曲折があり、80 年代の終わりから 90 年代の初めにかけて、要求をちゃんと捉えないとソフトウェア工学として不十分だということになった。要求工学という言葉はあるが、どう見てもそれは外側であると言いたい。

では、モデリングは何をやっているかというと、これは何を作るかを決める、決めなければいけない。ということは、Known Unknown、ここがわからないということをまず知って、そこのわからない部分を Known Known にする。不確かさを軽減する。抜けがあると後工程で変なことが起こって手戻りが生じる。不確かさを軽減する方法として何があるかというと、伝統的なソフトウェア工学では仕様を定義する。仕様を自然言語や形式言語で書いて決めていくのが普通である。

そうでなく実験ベースの方法もある。人間は最初から全てのことをわかって、物事を明文化していけるわけではないということ。特に外界とのやりとりがあるシステムというのは、どうしても仕様ベースではすまないことが多い。図 2.2.4 で、（ａ）は、割と簡単な組み込みシステムである。何を作るかということが、要求仕様が仕様書ベースで書かれていて、それをもとに作り、作ったものを検査する。設計レベルで検査する場合もあり、プログラムレベルで検査する場合もある。検査の一つの観点は入力の網羅性である。実際に実行したときに、「外側から変なデータが入ってきたらコア吐きました（システムクラッシュのこと）」では困る。よって、網羅的に物事を考えて、さまざまな入力の組み合わせに対し

て、システムが正しく意図通りに動くということを検査する。そのための技法のひとつがモデルチェッキングという技法である。

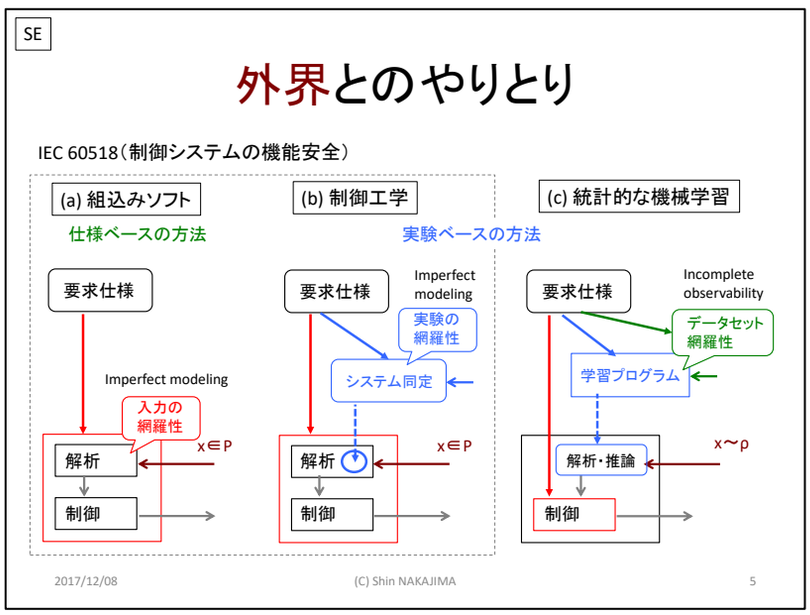


図 2.2.4　［SE］外界とのやりとり

でも、組み込みシステムはこういう簡単なものだけじゃない。もともと、組み込みシステムは、コンピューティングとは別のディシプリンとしての制御工学と関係が深い。したがって、どちらも IEC 60518（制御システムの機能安全）の対象物である。制御工学の場合には、制御対象が余りにも複雑なために、仕様が書けないことがある。小学生が使うような簡単なロボットだと古典力学でその振る舞いは書けるかもしれないが、化学プラントは書けない。その場合には実験をする。実験して、ある入力を入れたら出力がどうなるかという現象論的な方程式の簡単なものを作って、そこをつなぐ。実験ベースの方法である。それを制御工学ではシステム同定と呼ぶ。システム同定の結果として得られたモデルを使って実際に実行系を動かす。このときの網羅性は実験の網羅性となる。実験で抜け落ちがあるとだめ。

さらに、（ｃ）の統計的な機械学習の場合は、システム同定に相当するのは学習プログラムである。そこでの網羅性は、どういう訓練データに対して学習したか、つまりデータセットの網羅性になる。では、（ｂ）と（ｃ）は同じなのか。違う。何が違うかということを考えると、どうも不確かさの本質が違うのではないかという気がする。

不確かさというのは、「決まらない／決められない」こと。これは「決めない」とは違う。ソフトウェアの開発だと上流工程では決めない。プログラムにしていく段階で決めていく。例えば非決定性のあるデザインを書いておいて、それを実行可能なプログラムに落としていく。その過程で、決めないものを決める。

しかしながら、不確かさは決まらない／決められない（図 2.2.5）。「決まらない」の１番目は、量子力学で「神はサイコロを振る」、inherent uncertainty。２番目は情報が不完備、incomplete observability。結局、我々は全てを知ることはできない。３番目はモデリング

が不完全、imperfect modeling である。これは全てを考えると作れないということ。なので、どこかで適当なところでぽんと切る。これが実は今までやっているシステムの開発の方法である。

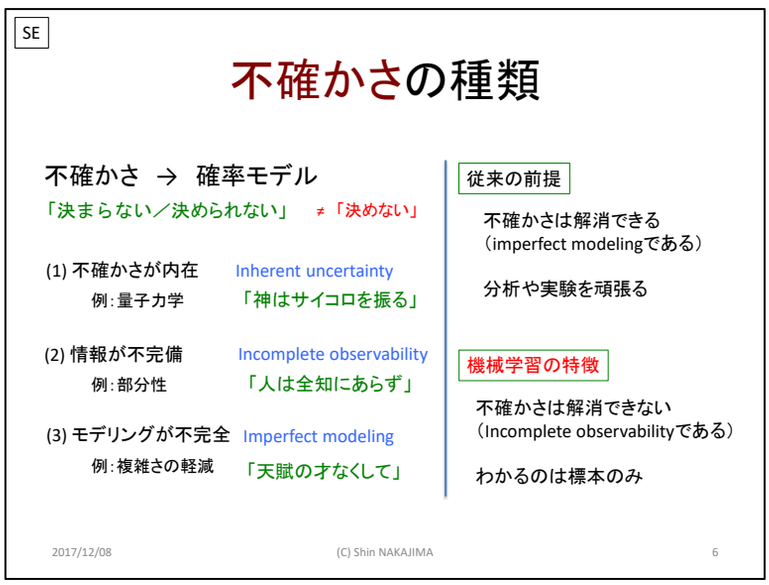


図 2.2.5 ［SE］不確かさ種類

不確かさに関する従来の前提は imperfect modeling である。頑張ればできるかもしれないけれども、さまざまなコストによってやらない、どこかで手を打っている。しかし、機械学習の特徴はそうではなくて、不確かさを解消できないのではないかと思う。incomplete observability。別の言葉でいうと分かるのは標本のみだということ。標本はわかっても、それの裏に隠れている確率分布はわかりようがない。統計学というのは多くの場合、隠れている確率分布はわかるということが暗黙の前提である。この点が大きな違う。不確かさが違うというところが機械学習の難しさの理由であると思う。

今度はソフトウェア工学の観点から機械学習を見る。不具合の原因について、ソフトウェア工学、特に IEC 60518、機能安全の規格では、ハードウェアは偶発的故障だけれどもソフトウェアは系統的故障だという。もちろん、ハードウェアにも系統的故障はある。しかしソフトウェアには偶発的故障はないという意味。ここで系統的故障とは、設計上の欠陥であって開発過程で混入したバグが原因となる故障のことである。だから、ソフトウェア開発のプロセス品質を頑張る。一部では、それがソフトウェア工学のことだと思われている。

では、機械学習の故障は偶発的故障なのか、系統的故障なのか。機械学習は丸山さんの発表（２．１）にもある通りソフトウェアなので、系統的故障である。では、デバッグしなければいけない。でも、実行推論結果は確定的なものではない。なんとなくどこかに偶発的な要素が入っているように見える。偶発的故障もあるかもしれない。それで、僕は機械学習のプロではないが、一生懸命、考えたのがこの絵である（図 2.2.6）。

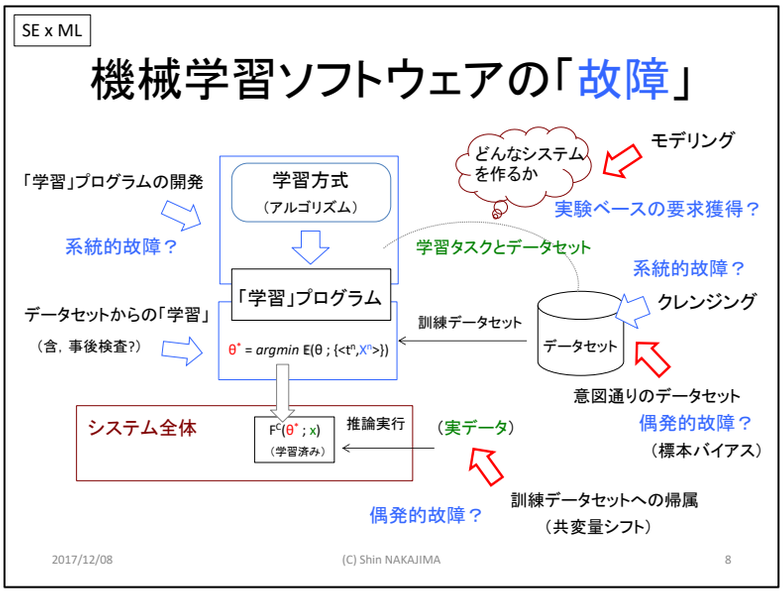


図 2.2.6　［SE×ML］機械学習ソフトウェアの「故障」

さまざまなコンポーネントを使ってシステムを作っていくが、そのときに、もしかしたら、ここの部分は系統的故障が入る可能性があるのではないか、ここは偶発的故障と考えたほうがよいのではないかという分類をした。学習プログラムの開発は系統的故障。そもそも何を作るかというところは、仕様書をがりがり書いて決めるものではないので実験ベースでやるしかない。プルーフ・オブ・コンセプトをやって、実験的に何をやりたいかということを整理していくという話かもしれない。データセットを作るところというのはクレンジング等、これらに系統的故障が入る可能性があるかもしれないし、それ以外のところには偶発的故障が入るかもしれない。よって、ハードウェア的な故障の考え方とソフトウェア的な故障の考え方の両方が入っているのかもしれない。

僕は機械学習プログラムのテスティングに関する研究をやっている。例えば、サポートベクターマシンの Java のプログラムとか、ニューラルネットワークの Python プログラムとか、それをテストしたいということである。テストというのは、テストデータを与えてプログラムを動かして意図通りの結果が出るかを調べることである。そのときにいろいろなデータを与えて、コーナーケース・テスティングをしたい。機械学習のプログラムは、一つのデータを入力するのではなく、データセットを入力するわけだから、データセットの中のデータの分布のようなものが影響する。それを、データセット多様性と名付け、幾つかの例題について実験した。時間の関係で事例紹介は飛ばす[4]。

ソフトウェア工学に戻るが、教科書的なV字でやっているわけではなく、開発方法というのは使い分けをしている。問題が良くわからないときには exploratory programming スタイルで、わかっているときには、理想的には、correct by construction で開発する。

[4] 時間の関係で省略されたが、詳細は下記論文に書かれている。
中島震「データセット多様性のソフトウェア・テスティング」（日本ソフトウェア科学会第 34 回大会講演論文集、2017 年 9 月）　http://jssst.or.jp/files/user/taikai/2017/FOSE/fose3-2.pdf

特に要求モデリングについて、IWSSD というワークショップで要求工学という言葉ができる前に議論した（図 2.2.7）。最初にスカルセッションをやる。何を作るかということ、スカルとは骨格のことで、システムの骨格をみんなで作っていきましょうと、インタラクティブなプロセスである。その後、要求の形式化をしますという話。こういうことを議論していた人は考えているわけだが、ソフトウェア工学の教科書には書いてない。機械学習でも、こういうスカルセッション的な部分をきちんとやっていけば、30 年も前からソフトウェア工学で議論している中のある部分は使えるではないか。

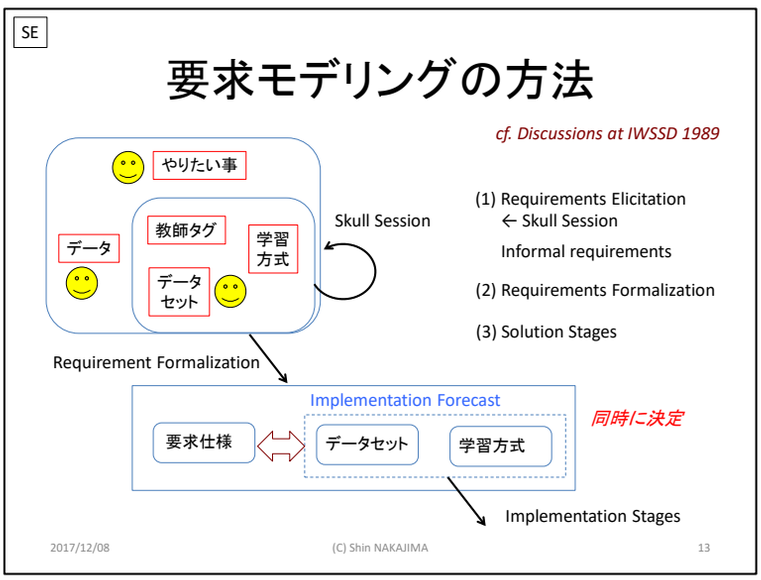


図 2.2.7　［SE］要求モデリングの方法

そのように考えていくと、SE×ML というのはさまざまな分野を総合的にやらなければいけない気がする。機械学習の基礎的な研究についても、先ほどの騙せる標本の例（誤認識）もあるし、説明性の話もある。デバッグやシステムの開発に関して、ハードウェア、ソフトウェア、機械学習、それらを全体として考えなければいけないと思う（図 2.2.8）。

最後に、これまでの話題とニュアンスが違うかもしれないが第三者評価について話す。例えば、セキュリティの分野ではコモンクライテリア（CC）というのがある。これはシステムのプロバイダとユーザ、それ以外に第三者の評価者がいて、ある程度、客観的にこのシステムは安心して使えますよということを言う。そのときに重要なのは、ここに書き忘れているが、レベル分けしているということである。機械学習を使うにしても、ミッションクリティカルなものからエンタメ系までさまざまにある。それごとに要求される品質の基準は違うので、そういうレベルをきちんと分けることで、利用者が安心して使えるようにしていく。それはシステム提供者側だけが、これは安全ですよと言って終わりになるものではなくて、きちんと第三者、中立機関がやるべき話であると思う（図 2.2.9）。

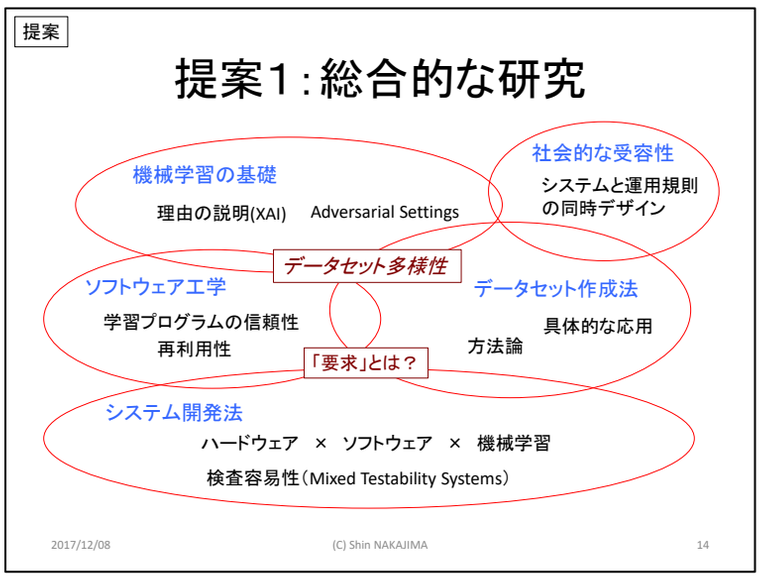


図 2.2.8　提案 1：総合的な研究

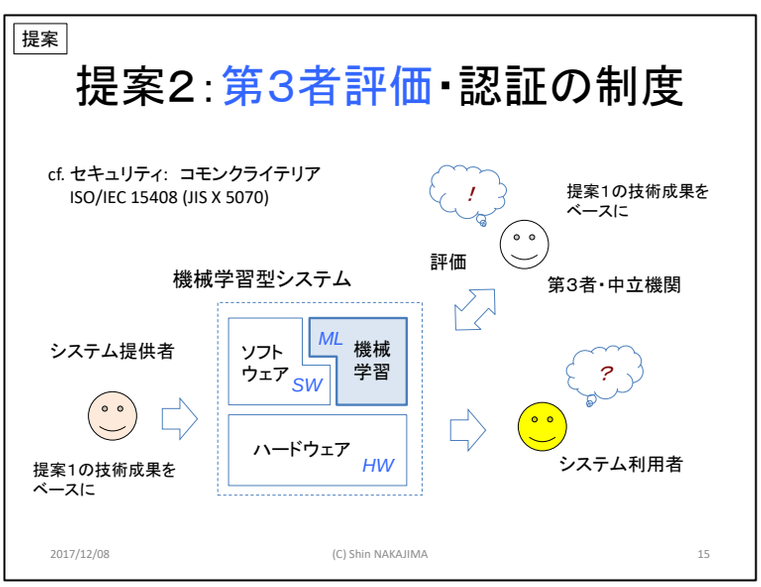


図 2.2.9　提案 2：第三者評価・認証の制度

【質疑応答】

Q：第三者評価に関する日本や世界の取り組み状況は？

A：産総研の関口智嗣さん（情報・人間工学領域長）から、いち早く日本が取り組むことで世界をリードできる可能性があると聞いた。なぜなら、コモンクライテリアについては、早く取り組んだ海外、特にフランスの評価機関は非常によいと聞く。

Q：不具合の原因のスライドで、「実行（推論）結果は確定的でない」とはどういう意味か？同じものを入れれば同じものが出るという意味では確定的では？

A：そういう言葉であればそうだが、訓練データセットが確定的に決まらないので、要求が正しく訓練データセットに表されているかということが問題になる。

Q：CPS について Tight Closed Loop と言っているが、タイトなループであることが必要か。オープンでルーズなループも CPS にはあり得ると思う。例えば、Industrie4.0 の文脈でいうと、もっと大きな考える工場というような。

A：あってもいいが、ここで強調したかったのは、CPS があまりにも組込みシステムの高度化にこだわりすぎているのが問題であるということ。CPS における基本的な見方はクローズドループであり、それが 1 個でなく小さなものがたくさん集まり合ってグループになって、その結果、全体の挙動が見えなくなるという。これが CPS 的なものの一番難しいところである。同じ階層レベルでつながっていて、ちょっとした擾乱がどこに波及するかどうかは自明ではないし、それが因果ループとして論じられていることが問題だと思う。

【補足：ワークショップ後に発表者から送られてきたメモ】

統計的な機械学習のディペンダビリティを考える背景

1982 年 ACM が『複雑かつ大規模なシステムを構築できるようになったことが、逆に、そのシステムの信頼性を怪しくし、社会に大きなリスクをもたらす可能性が増える』と危機感を訴えた。1984 年に『コンピュータ・システムに誤りがないという神話と反対に、現実には不具合に陥る。その信頼性を担保することができない。完全に不具合を取り除くことは不可能であるが、リスクを許容レベルにとどめることができると信じて、研究を進める』ことを提言した。

1985 年 SRI の P. Neumann が、同じ失敗を繰り返さないように情報共有することを目的に、コンピュータ・システムの不具合事例を収集し公開するリスク・フォーラムを開始した。ACM 学会誌 50 周年記念号に寄稿したエッセイは、コンピュータ・システムを取り巻く環境が大きく変化した結果、従来なかった新しいリスクが増えている状況を踏まえて、『リスクの実際を知る事で対応策を整理し、リスクを低減する技術の研究が着実に進んできたが、障害事例は増加する一方である』と論じた。新しい情報技術が広がるペースが速く、リスクを低減する技術は『一歩出遅れ』る。このことが、障害事例が増加する理由である。

2007 年科学アカデミー（NAS）は「ディペダブルなシステムのソフトウェア」と題するレポートを公開した。リスク・フォーラムの対象はコンピュータ・システム全般だった。ソフトウェアの比重が高まり、ディペンダビリティがハードウェア中心の見方である耐故障から信頼性と安全性に広がった。ソフトウェアがディペンダブルなことを示す証明書（certificate）が求められている。障害の事例と最善の対処法を収集し現状を把握すること、リスク低減に関わる基礎研究を進めることを提言した。（注：ちょうど同じ頃、CPS という言葉と共に、ソフトウェアが新しい方向に進み始める。深層学習が成功したのも、この頃だった。）

C. Perrow は、前記の NAS レポート委員会メンバでもあった社会科学者で、1984 年に『当たり前の事故（Normal Accidents）』という見方を提案した。破滅的な結果をもたらす事故は、些細なことから起こるべきして起こるというもの。この考え方に基づき、2011 年に『産業システムの中には破滅的な事故が起こる可能性を除去できないものがある』と指摘し、『適切な規制を導入しても破滅的な事故を避けられないシステムを作ってはならない』と論じた。

今、統計的な機械学習の技術が新しい可能性を拓くとして、大きな期待を集めている。P. Neumann の小論にあるように、リスクを低減する技術の研究開発を同時に進めていくことが肝要だろう。C. Perrow が指摘するような破滅的な事故のリスクが機械学習に残るとするならば、安心して使えるとは言えない。

安心して使うという言葉からイメージするのは、技術利用側のことかもしれない。ところが、絶対的にディペンダブルなシステムはあり得ない。新しい技術はチャンスと危険が隣り合わせにある。不具合の発生により、技術提供側のビジネスに大きな危機がもたらされる可能性がある。提供したシステムから得られるプロフィットに比べて、これを大きく上回るようなコストは引き受けられない。リスクが大きい場合、ビジネスが成り立たない。つまり、破滅的な事故は、利用側と提供側の両方から、「安心」を遠ざける。

安心さを得る方法のひとつは、利用側と提供側から独立した中立な第三者機関による「お墨付き」を与えることだろう。実は、これが NAS レポートの要点だった。つまり、第三者機関の根拠となる適切な規制を導入することが必要であるとした。技術一辺倒ではなく、C. Perrow が云う『使ってはならない技術』にならないような努力を、社会全体として積み重ねることである。

第三者評価の基本となるのは、評価者の技術力である。聞いたところでは、CC について、評価機関によってセキュリティ脆弱性を評価する技術力に大きなバラツキがあるという。技術力の低い機関は書類審査程度に終わり、真の脆弱性評価とならない。

さて、機械学習システムの第三者評価を行おうとする時、期待されているレベルの技術が確立しているだろうか。機械学習の魅力的な使い方や応用の探索は興味深い。同時に、利用目的に見合うディペンダビリティを持つことを保証する技術の確立という視点をベースに、ソフトウェア工学と統計的な機械学習、双方からの基礎研究を進めることが期待される。

ソフトウェア工学の課題としては、要求形成の方法、検査（ソフトウェア・テスティング）の方法、データセット作成の方法、など。統計的な機械学習の課題としては、学習結果の堅固性（ロバスト性）に関わる諸問題、たとえば、敵対標本、毒化攻撃、標本選択バイアス（共変性シフト）、Generative Adversarial Network (GAN) 等であろうか。これらの研究成果を、ソフトウェア工学の道具として使える可能性がある。

残念ながら、これまで、機械学習とソフトウェア工学の研究分野間交流はみられなかった。機械学習は「新しいことの実現」に関心があり、一方、ソフトウェア工学は開発の生産性と「ディペンダビリティ」に関わる。二つの分野は、価値観の異なる文化に属する。二つの文化にある垣根を乗り越え、統計的な機械学習ソフトウェアのディペンダビリティ向上という目的を共有して、研究交流を進めることが大切である。

2.3 AIを活用した社会価値創造の取り組みと今後の課題

谷 幹也（日本電気株式会社）

NEC（日本電気株式会社）では社会課題の解決に向けて、実世界でセンシングした情報をサイバー世界で見える化、分析し、計画・最適化を行った後、実世界へのアクチュエーションにつなげるといった社会ソリューション事業に注力してきている（図 2.3.1）。その中で AI はサイバー空間において、人間の知的活動をコンピュータ化した技術とも表現できる。基本的には認識、予測、計画・最適化といった機能を、機械学習をベースにして実現することが AI の目的となる。

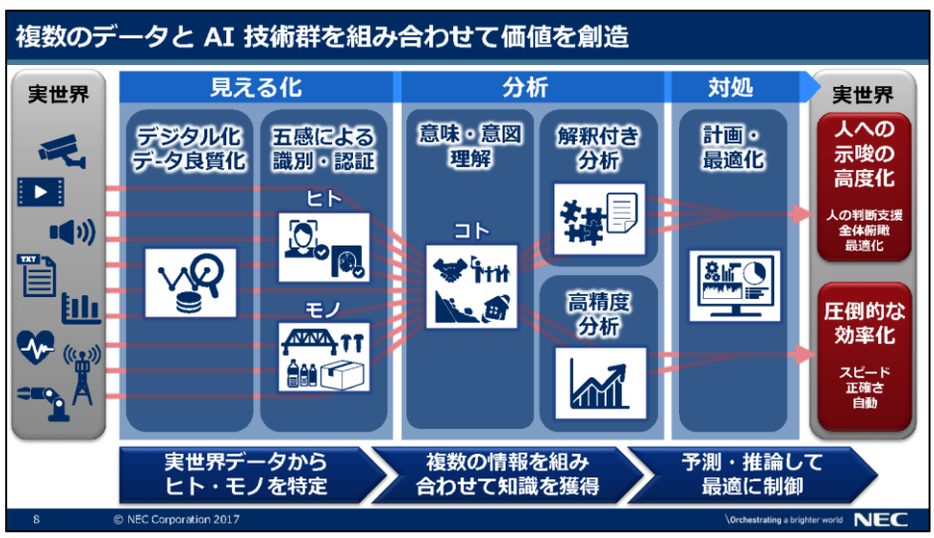


図 2.3.1　複数のデータと AI 技術群を組み合わせて価値を創造

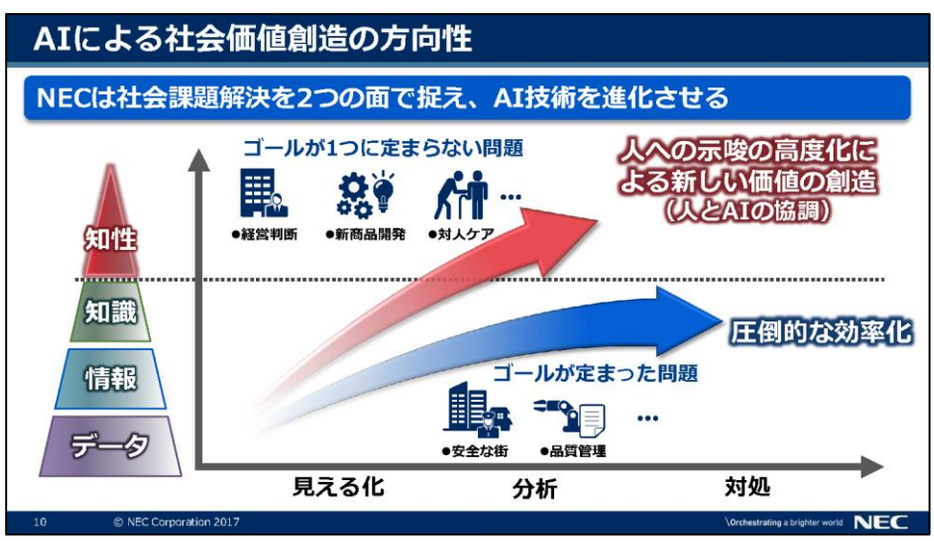


図 2.3.2　AI による社会価値創造の方向性

NEC が開発している AI 技術のほとんどは、見える化、つまり画像を見て人の顔であるとか、どのような物であるということを認識した後、それを解釈つきで分析して計画・最適化を行う部分と、高精度に早く分析して効率化する部分の二つに分けられる（図 2.3.2）。

前者は人と協調して解決するべき課題、例えば経営課題の判断や新商品開発、対人ケアなどの解決のために活用し、後者はアルファ碁のようなゴールが定まっている問題を超効率的に解決するような課題解決のために活用する。この二つの方向性で研究開発及び事業の開発を行っているが、そうした最先端 AI 技術群のことを「NEC the WISE」と呼んでいる。

実際に事業展開している事例としては、空港での入退場時の顔認識、インフラの劣化診断、工場のどこで故障が起こりそうだといった予兆分析、電力需要や商品需要の予測等がある。中でも機械学習がよく用いられている例としては、発電所における様々なセンサデータを取得して関連づけのところだけを学習した上で、そこでの相関関係から定常状態をモデル化し、それとの違いを捉えることで何かが起こりそうということを従来手法よりも圧倒的に早く検知するインバリアント分析がある（図 2.3.3）。また、出荷に対する在庫をどう持つかといった、非常に多くのパラメータがある場合を予測する際の異種混合学習技術といったものもある（図 2.3.4）。

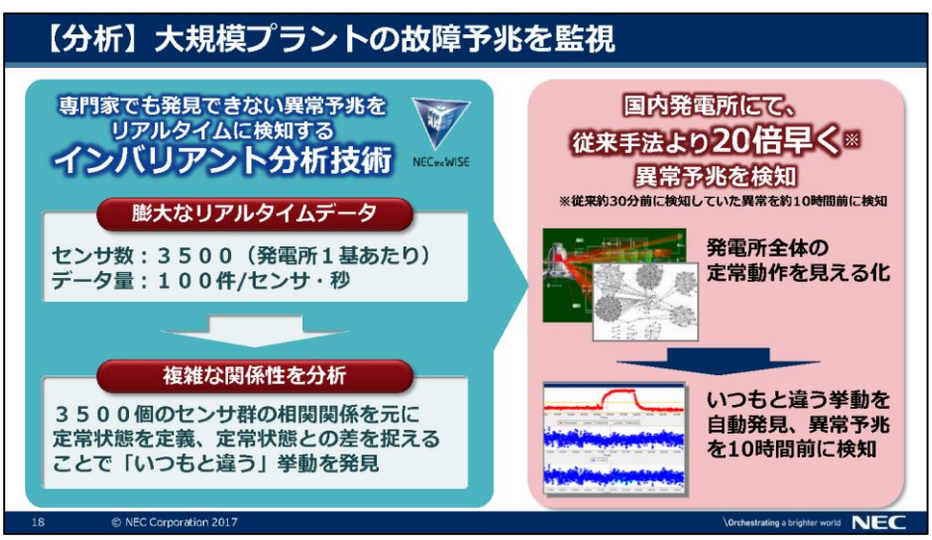


図 2.3.3　大規模プラントの故障予兆を監視

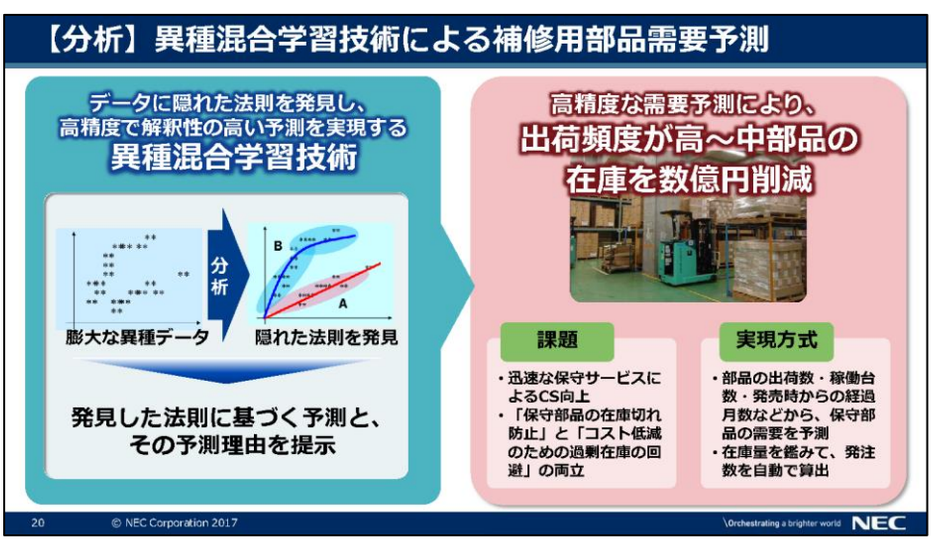


図 2.3.4　異種混合学習技術による補修用部品需要予測

しかし、結局、実際に AI を使ったビジネスをやろうとすると、まず、何をするのかということについてお客様と対話をしなければいけなくて、実はここが一番難しい部分になっている。なぜなら、お客様ご自身が正確に何をしたいか明確になっていないこともあり、とりあえずデータはあっても、これを使って何とかならないか、というところから始まることもあるからである。何とかというのは、結局、コスト削減か、売り上げ向上ということになる。そのどちらでもいいから、何か解を出してほしいぐらいの勢いで来られることも多い。お客様が本当は何をしたいのかということを最初に構想するところのコンサルから、検証、設計、導入、活用まで、実際にシステムとして動かすところまで含めてやっているというのが弊社の AI を活用したシステムインテグレーション（SI）事業となっている（図 2.3.5）。

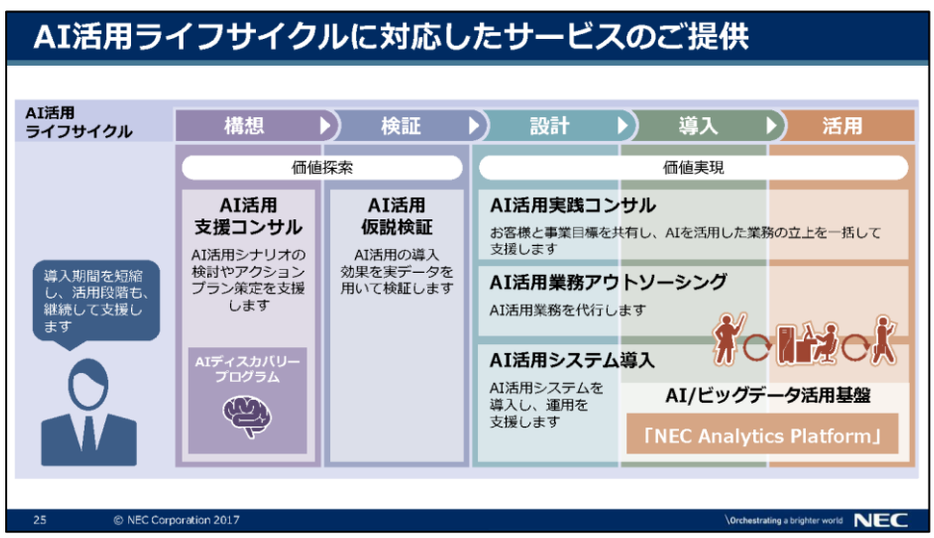


図 2.3.5　AI 活用ライフサイクルに対応したサービスの提供

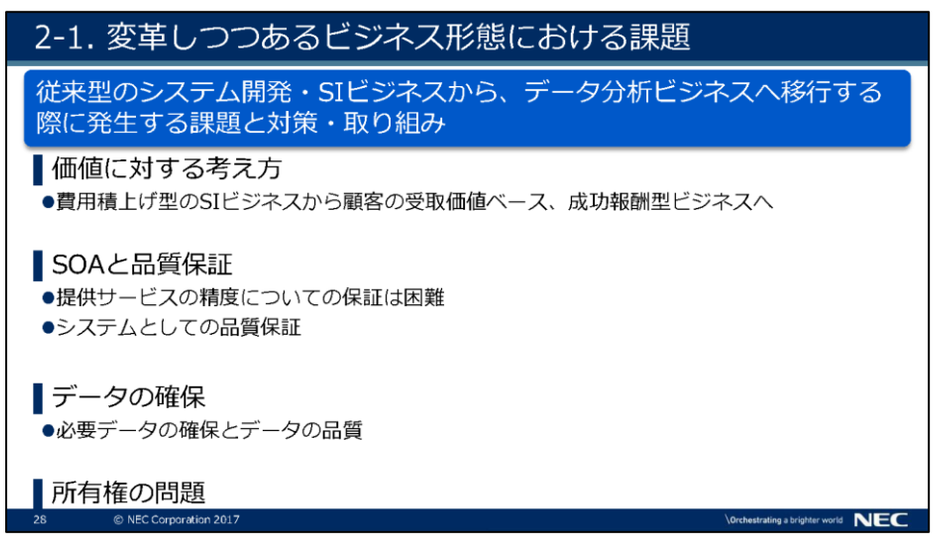


図 2.3.6　変革しつつあるビジネス形態における課題

さて、ここからが本日の本題となるが、従来型のシステム開発・SI ビジネスから、AI を含むデータ分析ビジネス形態に移るときにはどのような課題があるだろうか。図 2.3.6

に四つの例を示した。第一にあげられるのが、価値に対する考え方である。現行の SI ビジネスでは、そこにかかった原価に利益を乗せたものがお客様からいただく対価となる。一方、データを起点とした AI システムにおいては、お客様が認知されていない課題や解法を提示し、それを実際に解決するシステム/サービスを提供することになるため、お客様がその課題解決によって得られた価値を起点にその対価を決定するというようなビジネスの形態となっても良いと考えられる。ただ、現在までの商慣行上、この考え方が受け入れられないことも多い。

第二に、構築したシステムの品質保証 あるいは 検収に関しての課題がある。例えば、品質保証の話で言えば二つある。提供サービスの精度がどのくらいあるか、つまり、何%ぐらいで予測できるのか、99.99%なのか、といった予測精度についての保証というのは難しい。また、システム自体の安定性、品質保証については、機械学習の学習済モデルの動きを分析できていないことから、既存システムを分析して比較するところの精度の保証まではできないというのが現状である。

第三に、データの確保の問題がある。お客様としては、データを出したくないが分析はしたいという場合が多く、そうした場合はお客様のサイトで行なう必要があり、NEC 内のシステムを十全に使ってできない場合もある。また、実際にデータを見てみると前処理をかなりしないと使えない内容のものやデータの欠落があり、他のデータで補完する必要のあるものが結構多い。従って、必要とされる高品質なデータの確保が重要な課題となる。

第四に、出来上がった学習済モデルの利用に関しての問題がある。これは契約上の大きな課題でもあるが、弊社のデータに基づいて作ったシステムを他に転用してはいけないとか、あるいはその AI の部分を使うのであれば、ライセンスを渡してほしいといった問題から始まる。契約形態の問題だと言ってしまえばそうだが、そこの所有権を AI システム提供側で持つ契約をしようとすると難渋する。

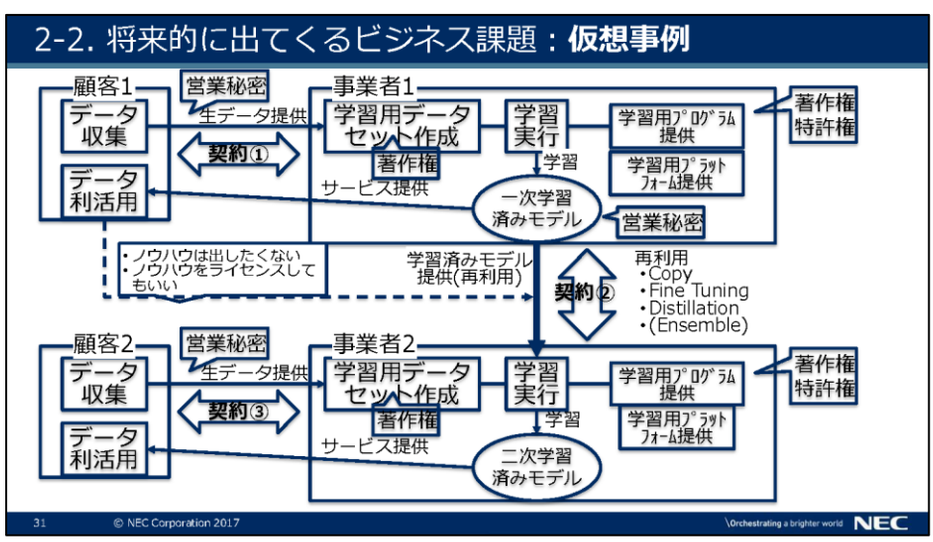


図 2.3.7　将来的に出てくるビジネス課題：仮想事例

将来的に出てくるビジネス課題としては、図 2.3.7 のようにデータ収集者、学習用データセット作成者、学習用プログラム提供者、学習プラットフォーム提供者、運用者といっ

た、ステークホルダーが多い状況への対応が考えられる。再利用としてのコピー、ファインチューニング、ディスティレーション、アンサンブルの使い方や、それらに対する契約問題、生データ及び学習データセットそれぞれに著作権、特許権等々が発生するのかしないのか、さらには一次学習済みモデルを二次学習のところに持っていって使うときに、どうライセンス料を発生させるのか、等々の問題が発生する。ディスティレーションは模倣だがコピーではないと言われるときに、どうやってお金をとるかというのも課題である。

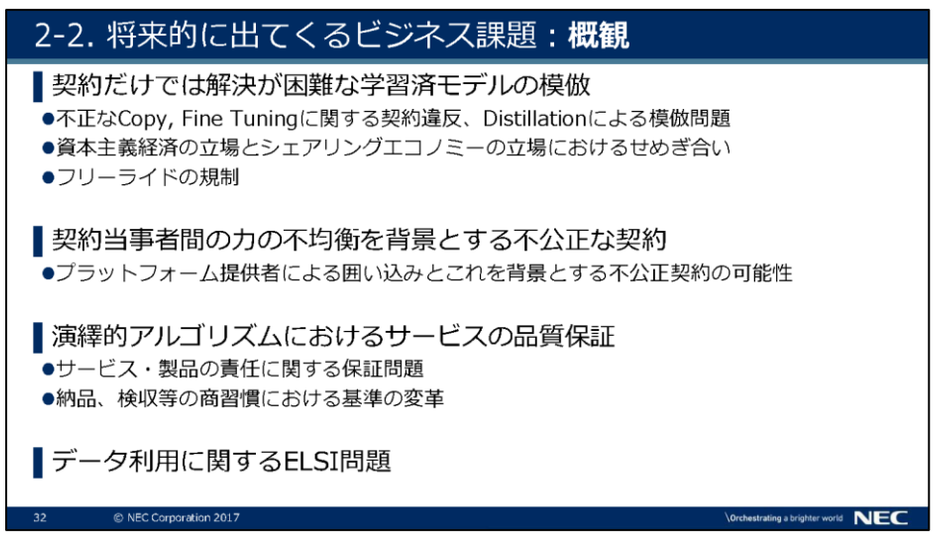


図 2.3.8　将来的に出てくるビジネス課題：概観

こうした内容を図 2.3.8 にまとめている。一番上にある、契約だけでは解決が困難な学習済みモデルの摸倣というのは、今お話したこと。不正コピー、ファインチューニング、ディスティレーションによる模倣といったことは実はやっていても分からないのではないかということがある。デジタル著作権管理（DRM）の研究をしていたときからこれは問題だったので、そう簡単に解決できる問題ではない。次に、資本主義経済の立場とシェアリングエコノミーの立場におけるせめぎ合いと書いているが、どちらかと言えば、シェアリングエコノミーの中で AI はコモンズになっていくのではないか。そうすると、どんどんオープンソースソフトウェアになっていって資本主義が成り立たなくなるのではないか、というところまで含めて考えなければいけないのかもしれない。それと、勝手に使うだけ使ってしまう人たちに対するフリーライドの規制の問題もある。

その次に、プラットフォーム提供者による囲い込みとこれを背景とする不公正契約の可能性と書いた。例えば、利用者は Google の TensorFlow 上でプログラムを一旦書いてしまうと、通常、なかなか乗りかえられなくなるが、その後 Google がライセンス契約を変えてしまっても TensorFlow を利用し続けざるを得ないといった問題がある。

その他にも演繹アルゴリズムについてのサービスの品質保証やデータ利用に関する ELSI の問題がある。私たちが実世界から集めてくる情報には、パーソナル情報が多く含まれているので、プライバシーを守りながら AI をつくらなければいけない。しかし、その AI はそうした情報を元に出来上がったものなので、ここにデータを逆に入れていくと、パーソナル情報が一部復元できるというようなことが知られている。こうした部分がきち

んと漏れないようにしなければいけない。また、AI が人に代わって高度な制御を行うようになった場合の安全をどう担保するかということもある。これは PL 法や機能安全の問題と関わってくるが、例えば自動運転のプログラムが間違っていたとき、その自動車メーカーはどこまでの責任を負うか、といったことがある。

もっと将来的な話では、AI によって人が代替された時、人はどういう仕事をすればよいのか、また、新しいビジネスモデルが生まれて新しい職業が誕生するといった話もある。私たちが今、注目しているセキュリティのところでは、現在の時点で日本では人材が 7 万人足らないと言われているが、そうした人材を育成したとしても AI によって代替されていった場合、余った人をどうするかといった問題も起こり得る。こうした社会的な課題に対しては、AI に対応した法整備、制度設計も行っていく必要がある。

NEC における人材育成においては、図 2.3.9 に示しているようにいくつかのデジタルトランスフォーメーションの専門部隊を強化してきており、2018 年度中には 1,000 人体制にしようとしている。この中でも特にデータサイエンティストとして研究者を採用していこうとしており、社外、大学からふさわしい人材に入ってきていただこうと取り組みを進めている。もちろん社内での人材育成もきちんとできるように、NEC の中で考えるスキルセットと、そのスキルセットに対する研修制度を構築している。社内育成の人材教育のフレームをつくり、他の職種、例えば SE や光関係から移ってくる社員を再教育して、活用、昇進させていく制度も整えてきている。

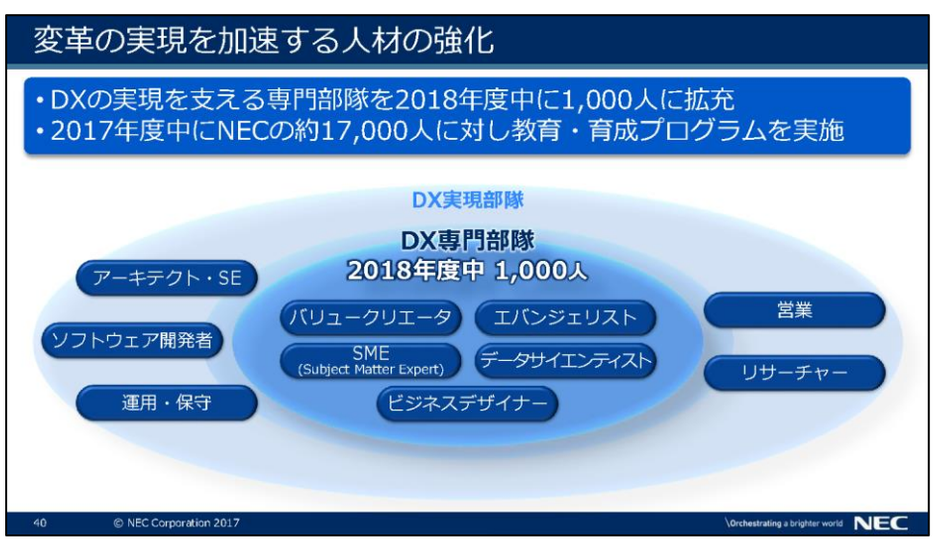


図 2.3.9　変革の実現を加速する人材の強化

【質疑応答】

Q：価値に対する考え方と関連するが、従来の費用積上げ型の SI ビジネスから成功報酬型ビジネスに移行させていくことに対して、顧客はどのくらい納得してくれるか？

A：納得していただくのはかなり難しい。ただし、今までできなかったことができるようになるといった場合にはある程度、支払っていただいている。原発の例で言えば、一旦シャットダウンして問題点を洗い出し、それからまたリスタートしようとすると膨

大な費用がかかるが、シャットダウンしなくても問題点を発見できるといった場合には、大きな価値があると理解していただきやすい。このような決定的に重要な場面ではうまくいく。しかし、今までもできていたことでコストが 2 分の 1 ぐらいになるといったところに関して言えば、その 2 分の 1 の部分の何%でもいいから支払ってほしいと言っても、なかなか納得していただけないところがある。また、契約のところでかなりもめるというのと、契約するときにはまだわかっていないということが厄介な問題になっている。例えば、最初、概念実証（PoC: Proof of Concept）をやってみるという契約を立てて実際に PoC を実現して見せて、納得していただいた上で次にいくというステップ・バイ・ステップでビジネスをやらざるを得ないが、この 2 段階目の交渉がなかなかうまくいかない。こうしたことから PoC ばかりで何やっているんだ、と言われもするが、PoC をやってみないと本当に役に立つかどうかが分からない。お客様がコストダウンを 30%実現したいと言っている問題に対して、私たちのソリューションを使えばそれ以上の 60%ぐらい確実にコストダウンができるといった場合には価値を認めていただける。しかし、30%のコストダウンを要求されているところで、その通りに 30%と言ったら、それは普通にやるでしょう、といった議論に巻き込まれることもあってなかなか難しい。

Q：今の話は日本企業だからという側面はあるか？　海外でも同じか？

A：NEC はグローバルな企業ではあるが、国内に限らず海外でも PoC までで終わっていることが多いと聞いている。ただし、実ビジネスまで行き着いたものでは認めていただいている場合もある。一方 GE は、今までの燃料効率からこれだけ下がりました、といった事実に基づいて何%というような支払いがなされているのでうまくやっていると思う。

C：そうした例を結構耳にするので質問した。GE のような契約が日本企業では難しいというのは、日本企業のメンタリティに依存する部分があるのかなと思った。

A：そうした部分も大きいと思う。

Q：AI 的な手法が広がっていったときに、どのような領域、業務に AI、機械学習が適用されていくことになるか？　また、そのときに起こり得る品質保証等の問題点がどの辺で顕在化することになるか？

A：まずは、人間が全部を見て判断できないぐらい大規模なシステムへの適用があるのではないか。人間がこなせないぐらいのスピードか、人間が見切れないぐらいの大規模性、複雑性というところのシステムはありだと思う。

Q：監視等もそうしたニュアンスがあるのではないか？　人間はモニター100 台の全部を見ることができない。AI で怪しいところだけを抽出して、その後の判断は人間に任せるというやり方。

A：その通りだと思う。監視のところは、スマートシティの検討の中で私たちもやっている。ただ、その中で顕在化してきたこととして、そこでのスピードというボトルネックが、アクチュエーションの方になってしまうことがある。つまり、何かが起こったときに、そこに警察や警備の人が行くという最後のところのパスが早くないと、幾ら監視のスピードが上がったとしても変わらないと言われてしまう。例えば、最後のア

クチュエーションまで自動的にできる場合の一つとして、海外での高速道路についての事例がある。シンガポールでは大雨が降ったときにレーンに薄い水がたまるため、高速道路のレーンをうまく切り替える必要があり、この時はアクチュエーションまで自動化できて全体の速度をあげることができた。ただ、例えばサーベイランス、監視のところで、今行われている犯罪を制止しないといけないというミッションが入るとなかなか難しい。空港のゲートみたいに、ゲートがあって必ずここで止めるといったものがあればいいが、オープンな空間で監視をするというときにはなかなか難しい。もちろん、警備会社と組んで、そうしたところのスピードをうまく上げるような方法が出てくればありだとは思うが。

Q：NEC のビジネスの中で、学習したネットワークモデルといったものを顧客に渡すときに、著作権のような問題にはどのように対応しているのか？
A：今のところケースバイケースで対応している。最初に契約をどう結んだかが決定的なポイントになっているが、ガバナンスを効かせて契約を全部コントロールするというのは難しい。現場で結んでしまった契約があったりすると、そのまま全部をお渡ししなければいけなかったりというのはある。ただし、大型の案件は大体、全社でガバナンスをかけてやっているので、業務上のノウハウではない部分を他の業種に転用するときは、ある一定のところまでは良いと許可いただいている。
Q：その場合、派生モデルの扱いというのはどうなるのか？
A：そこまでいった案件が今のところまだない。各モデルは微妙に違っている。例えば原子力発電所のモデルというのも簡単ではなくて、加圧水型と沸騰水型で違ったり、あるいはそこでのモデルは火力発電所に持っていけなかったりする。学習の仕方で解決できるのではと思われるかもしれないが、実際には過学習等でうまくいかないところがある。
Q：顧客に提供するモデルの最初のベースを例えばアレックス・ネットからとってくるとか、そういうことはしないのか？
A：今はしていない。

Q：最後の方で人材、データサイエンティスト育成の話があったが、その場合の講師は誰がするのか？
A：社外の有識者の方にお願いする場合と社内のエキスパートで行う場合の両方がある。後者の場合は、既に上級のサイエンティストとして登録している者が何名かいるので、その人たちが講師となって進めている状況である。本来はそういうカリキュラムが NEC だけではなく、日本全体で定められたものとしてあると良い。それを共有化できればそこからさらにプラスアルファのこともできると思っている。

2．4 機械学習型システム開発へのパラダイム転換 －MPRG の研究紹介－

藤吉 弘亘（中部大学工学部ロボット理工学科）

我々、中部大学・機械知覚&ロボティクスグループ（MPRG）は、中京大学、三菱電機とともに、TEAM-MC2（エム・シー・スクエアード）というチームを作って、Amazon Robotics Challenge（ARC）に挑戦している。Amazon Picking Challenge と呼ばれていた初回の 2015 年シアトル大会から名称が ARC に変わった 2017 年名古屋大会（図 2.4.1）まで毎年参加しており、継続的に参加することによりさまざまな課題が分かってきた。本日は、その課題も踏まえて、MPRG の研究を紹介したい。

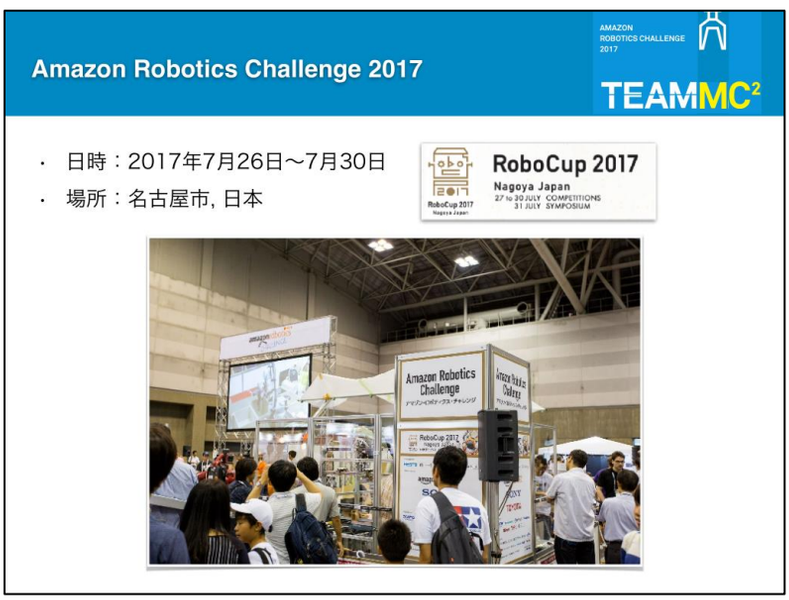


図 2.4.1　Amazon Robotics Challenge 2017

ARC は物流の自動化を競うコンペティションであり、棚に陳列されたアイテムの中から指定されたアイテムをロボットが認識し、ピッキングするとともに箱へストーイングする問題を扱っている。Amazon の倉庫内では、注文が入るとロボットが棚まで行って、棚を持ち上げて持って来るというシステムが既に稼働している。ただ、ロボットは棚を持ってきてくれるが、棚の中からアイテムをとるという動作は人間がやっている。Amazon はそこを自動化したいと考えているが、Amazon の子会社 Amazon Robotics でも自動化の実用化は難しい。そのため、Amazon は、それをオープンにして、しかも、国際的な競技会という形で 2015 年からコンペティションを開始した。2016 年までは、商品アイテム 40 種類すべてが参加チームに公開され、商品アイテムや 3D モデリングのデータが配布されていたが、2017 年からは、商品アイテムの事前公開を半分のみに限定し、残りの半分は競技の 45 分前になって参加チームにはじめて知らされる形式に変更された。これは、45 分で学習データをとってアノテーションして学習する厳しい作業であったが、棚入れ競技では、1 位の MIT-Princeton（マサチューセッツ工科大学&プリンストン大学）、2 位の Nanyang（南洋理工大学）についで、我々の TEAM-MC2 は 3 位の成績を収めることができた。

TEAM-MC2 では、基本的にはコンピュータ・ビジョンは我々中部大学が担当し、深層学習を使った。シングルショットアプローチと言われる SSD（Single Shot Multibox Detector）をベースとして、未知アイテムの物体検出ができるように、画像から特定のカテゴリの物体が存在する矩形領域（Bounding Box）を異なる解像度でも抽出するアプローチを取っている。大会に使われる 40 種類の商品アイテムの内、半分は未知アイテムなので、事前にそれ以外で違うようなものをいろいろ選び出した 97 種類を次々に取り替えて、11960 枚を学生が撮影して画像データを一生懸命作り、大変手間のかかるアノテーション作業を行った（図 2.4.2）。しかし、ビジネスにおける納期のように、競技する日が決まっており、商品が送られてくるのが 4 か月前なので、準備する期間が非常に短い。したがって、より良いものを作るためには、計算機パワーを借りて、いかに学習する時間を速くするかということが重要になってくる。今回は、NVIDIA Quadro P5000 という高速の GPU30 台を用意し、Chainer マルチノードと ChainerMN を使って、計算時間を 0.75 時間/epoch に大幅に時間を短縮することができた。

図 2.4.2　構築したデータベース

Amazon では、実際には、5 万点を超える商品を扱っており、毎日どんどん増えている。このような大量データを認識するためには、学習データをどうやって収集するか、たくさんデータを集めることもできないので、できる限り見ただけで学習できなければならない。データベースのスケーラビリティや大量学習データの収集コスト削減に関する問題が、ARC へのチャレンジから見えた課題であり、非常に重要な研究課題である（図 2.4.3）。

物流という世界では、商品のパッケージが日本用に変わるように、地域特有な性質がある。したがって、日本に特化したデータベースを作成し、その上で動く良い認識エンジンを深層学習も含めて機械学習ベースで作ることは重要な試みである。そこで、今、NEDO の委託事業「次世代ロボット中核技術開発」として、クラウド型の日用品データベースとその認識エンジンの提案を行っている（図 2.4.4）。商品を製造している企業から三次元的な CAD モデルをもらって、CG レンダリングによって多視点画像を生成して学習データに

利用することで、さまざまな多視点の映像、画像等を提供することを考えている。もしくは、日用品データベースと連動した認識エンジンを作成し、ロボット自体がダイレクトに認識エンジンとデータベースと連動して商品をピッキングするようなサービスを提供することもできる。

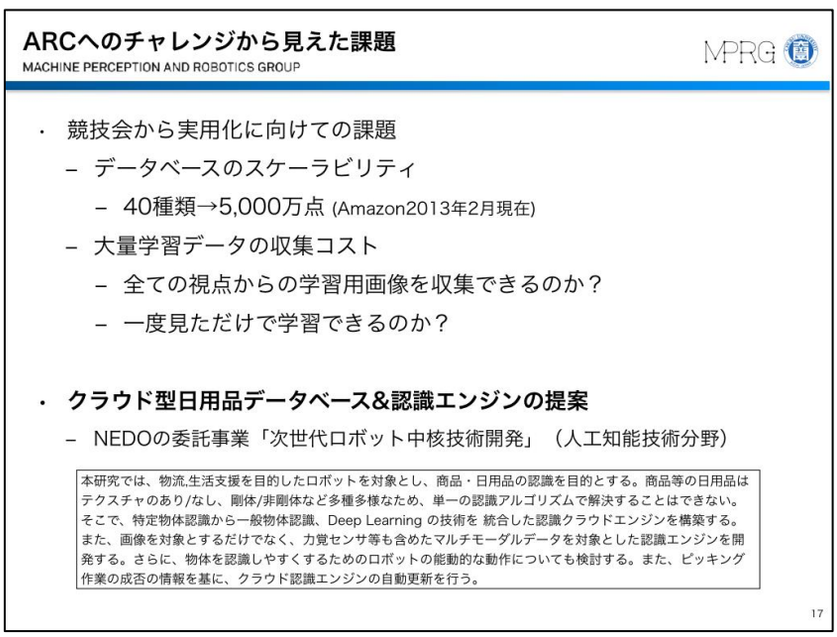


図 2.4.3　ARC へのチャレンジから見えた課題

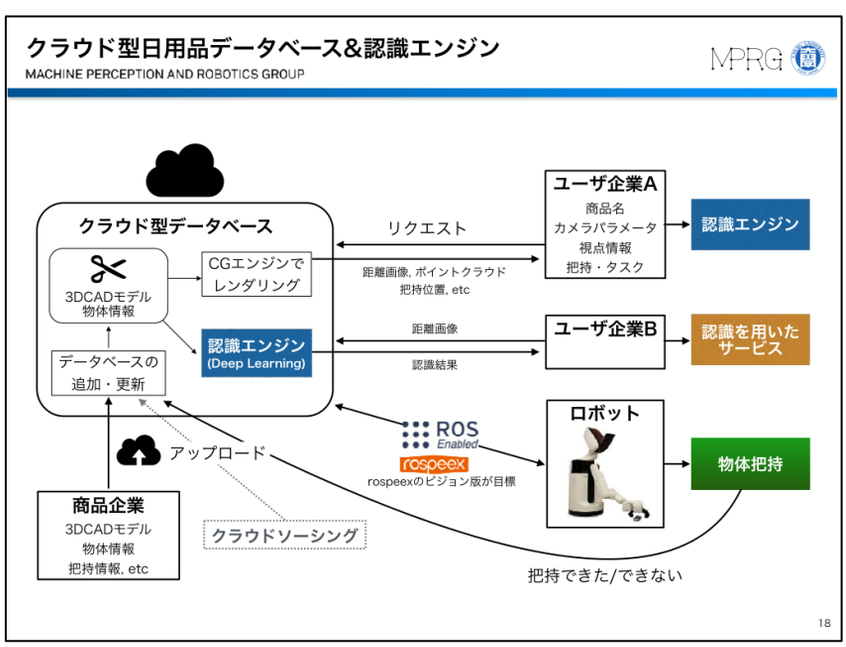


図 2.4.4　クラウド型日用品データベース&認識エンジン

未知の物体でも、ある程度似たようなものだと認識できるが、ロボットのハンドの形状によってもピッキング（把持）の仕方が変わるので、正確にピッキングするのは難しい。したがって、さまざまなロボットに試させて、試した結果をデータベースにまたフィード

バックして蓄積していけば、その後はどんどん一発で正確にピッキングできるようになる。また、我々は、深層学習におけるドメイン適応を目的として、少量の実画像と大量の CG 画像で学習することを試みている。今は日用品のデータベースとして、Amazon の課題に用いられているものを拝借しているが、このようなデータベースを作成してレンダリングすれば、CG ベースで疑似的な視点画像と距離画像が作れる。CG で作成したデータで学習して、実画像でうまくいけばコストが下がって、大変なアノテーション作業はなくなるが、そんな簡単にはいかない。なぜならば、CG のクオリティをどんどん上げてリアルに近づけようとすると、人的コストが非常に大きくなってしまうからである。そのために、ニューラルネットワークと敵対的訓練（adversarial training）を組み合わせることにより、実画像と CG 間の差異を吸収して CG らしさを差し引くような学習をして、ドメイン適応を試みている。

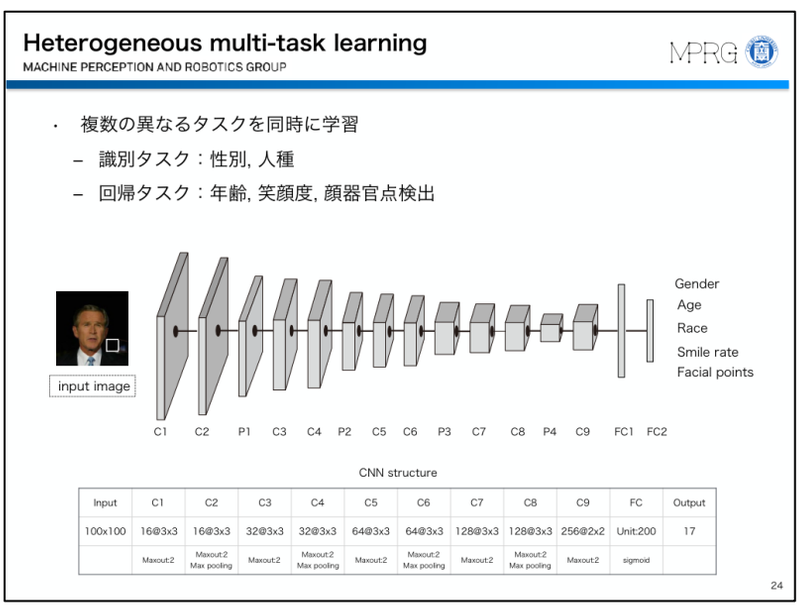


図 2.4.5　Heterogeneous multi-task learning

ロボットで動かすときに推論処理は非常にコストがかかるので、その推論処理をなるべく少なくしたい。そのために、単一のネットワークで複数のタスクを学習する Heterogenous multi-task learning を用いて、顔を検出した後に、性別、人種、年齢、顔器官点、および笑顔度を 1 個のネットワークで解くアプローチを採用している（図 2.4.5）。これは、今まではシーケンシャルに並べて行っていたものが同時にできるという意味で、推論の効率化につながるものである。

また、ロボットの計算リソースは限られているため、認識エンジンなど計算コストの高い処理を、クラウド上のコンピュータ行うクラウドロボティクスという概念への関心が高まっている。我々は、深層畳み込みニューラルネットワーク（Deep Convolutional Neural Network: DCNN）の処理をロボットとクラウドサーバで分割して処理するシステム（例：生活支援ロボットシステム）を提案している（図 2.4.6）。

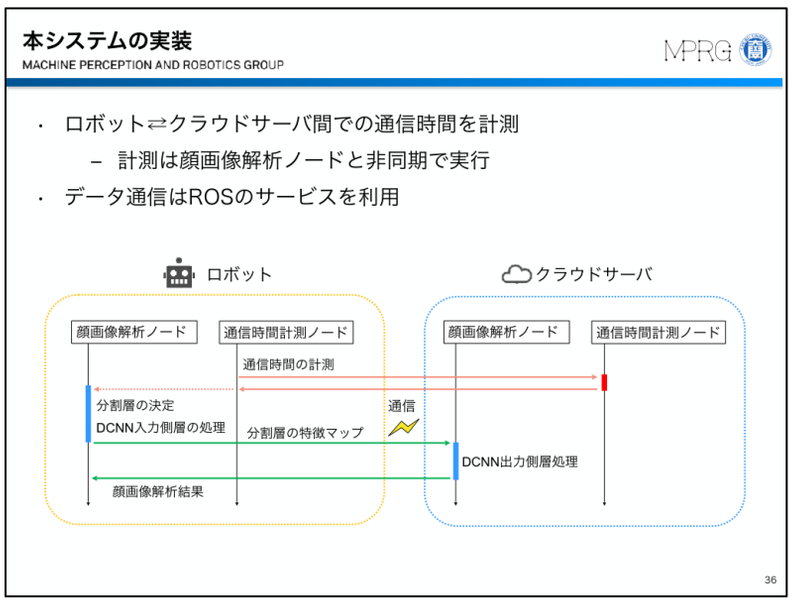


図 2.4.6　本システムの実装

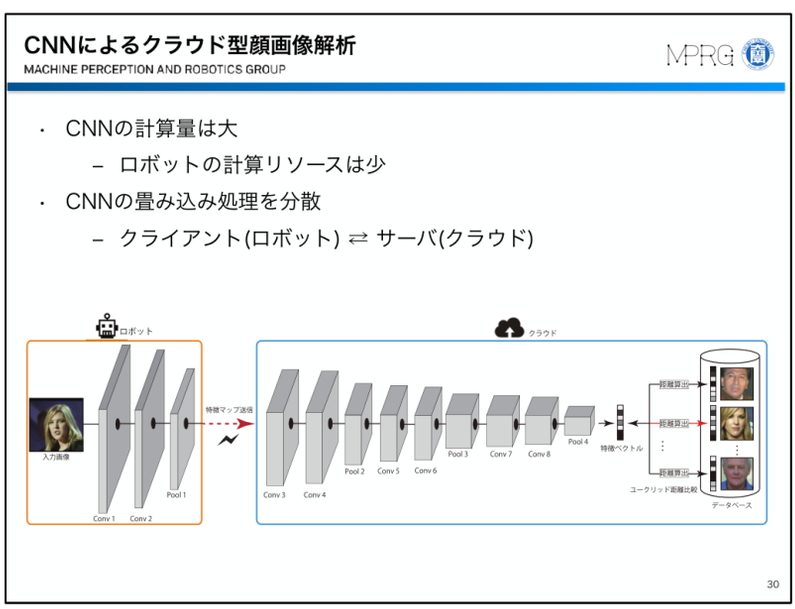


図 2.4.7　CNN によるクラウド型顔画像解析

ロボットの性能が高ければ、ロボット側でたくさん処理をして、残りはサーバーで処理する一方、ロボットの性能が低い場合にはなるべく少ない最初の浅いところの処理までをして、残りはサーバーで処理する。それをネットワークの負荷やロボット自体の性能を考慮して、動的に判断してやるような仕組みである。計測は顔画像解析ノードと非同期で実行し、データ通信は ROS（Robot Operating System）のサービスを利用している。分割した特徴マップをクラウドサーバに送信するので、DCNN 処理の一部をクラウドサーバ側で担うことが可能であり、送受信する特徴マップは個人を特定できる情報が削られているため、プライバシーに配慮することができる（図 2.4.7）。

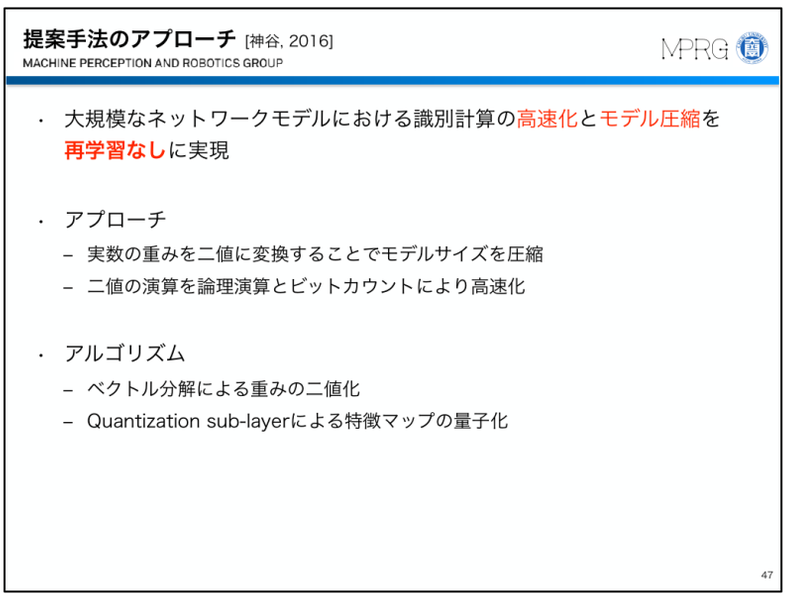
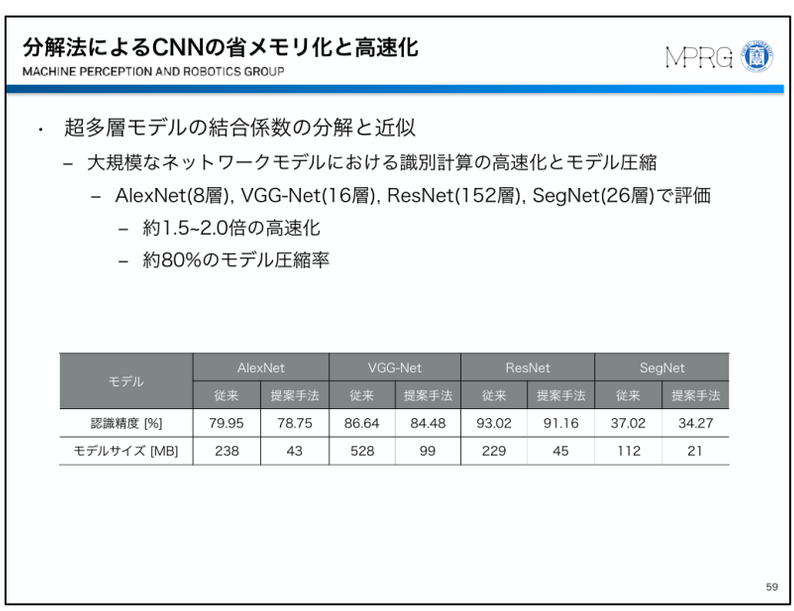


図 2.4.8　提案手法のアプローチ

最後に、ネットワークモデルの高速化に関する研究を紹介する。

画像中のシーンを理解する上で重要な認識タスクとしてセマンティックセグメンテーションがあり、最近では深層学習により解かれ始めている。畳み込みニューラルネットワーク（CNN）はより深くなっており、認識精度はよくなってきたが、低スペックの組込み機器では計算量の削減とモデルサイズの圧縮が必要である。ネットワークモデルの圧縮では、Deep Compression を用いるとネットワークモデルの再学習が必要になり、また XNOR-Net を用いると大規模なネットワークモデルでは識別性能が大きく低下してしまう。我々は、大規模なネットワークモデルにおける識別計算の高速化とモデル圧縮を再学習なしに実現する手法を提案している（図 2.4.8）。



モデル	AlexNet		VGG-Net		ResNet		SegNet	
	従来	提案手法	従来	提案手法	従来	提案手法	従来	提案手法
認識精度 [%]	79.95	78.75	86.64	84.48	93.02	91.16	37.02	34.27
モデルサイズ [MB]	238	43	528	99	229	45	112	21

図 2.4.9　分解法による CNN の省メモリ化と高速化

実数の重みを二値に変換することでモデルサイズを圧縮し、二値の演算を論理演算とビットカウントにより高速化するものである。ニューラルネットワークは入力ベクトルと結合係数（重み）ベクトルの内積計算だらけなので、内積計算を速くするために、全結合係数の重みベクトルにベクトル分解を適用するとともに、特徴マップの量子化と二値基底行列を用いた近似計算も行っている。これは CNN の省メモリ化と高速化につながり、AlexNet (8 層)、VGG-Net (16 層)、ResNet (152 層)、SegNet (26 層)の各モデルにおいて、1.5～2.0 倍程度の高速化と 80%のモデル圧縮率が達成できている（図 2.4.9)。

【質疑応答】

Q：思ったように精度が出ないときには、コードが悪いのか、データがいけないのか、あるいは前処理がいけないのか、その辺の切り分けに関するテクニックはあるか？

A：学生には、教育的観点から、最初にデータをちゃんと見るように指導している。そして、アノテーションを学生自身にやらせるようにしている。それは、アノテーションをすることによって、どういう変動があるかが分かるからだ。例えば、晴れの日のデータしか用いない場合、データには必ず影があるので、その影を機械学習で捉えるために、簡単に歩行者検出ができて精度が見かけ上 100%になってしまう。これは、重みベクトルのどの辺の特徴次元に重み付けがされているかということを見ることによって理解できる。したがって、データをアノテーションする際には、データを作る人、ネットワークモデルを設計する人が、まずデータをよく見るべきであると思う。アノテーションした後は、なかなか難しくて、上手くいった点、上手くいかなかった点をいろいろ見て、そこに何か共通性があるかないかということを推測しながら、やるほかはない。最近では、次元圧縮手法である t 分布型確率的近傍埋め込み（t-Distributed Stochastic Neighbor Embedding: t-SNE）を使って、高次元のデータを可視化し、分布から外れているところがどういうふうに外れているかを前もって見ることもしている。

Q：Amazon Robotics Challenge から見えた課題のところでおっしゃっていた、一度、見ただけで学習は可能なのか、技術的な方向性としてはどんなふうに考えているか？

A：深層学習には、ワンショット学習（one-shot learning）やゼロショット学習（zero-shot learning）という分野があり、研究者は多い。ただ、個人的にはこういったところで使うには、まだまだ技術は熟していないと思う。したがって、今回、我々は学習系ではなくて、いわゆる距離計算系でマッチングするようなアプローチで、当日配付されたアイテムに対処した。そのあたりをいかにこういった機械学習ベースできちんとクリアしていくかが課題である。

Q：まだ、これはという方式が出ているわけではないという状況か？

A：このような実用的な問題設定に合わせてやっているのはまだ少なく、企業の方と一緒に、そういった問題を解いていくということが重要である。基礎研究の域からは、まだ出ていない分野であるが、実用的に解くところは非常に大きな挑戦的課題である。

Q：例えば、40 種類の学習したものがあって、それに 41 個目を追加したときには、一番単純なのは 41 個でもう一回やり直すことだが、それはばかばかしい。40 個の認識タスクは既にできているわけだから、1 個目をどうするかというのにフォーカスした学習方式はどうなのか？

A：41 個全部を学習した方が良くなるので、まだまだ研究すべきところがたくさんある。

Q：三次元的にさまざまな角度で見る場合、機械学習ではさまざまな角度から見た画像を入力するが、人間ならば頭の中で回すことができるので違った学習ができていると思う。そのような観点からの研究はあるか？

A：人間は、与えられたものに対して、たぶん学習済みだから、それができるのだと思う。完全に与えられたものそのものでなくても、似たようなオブジェクトについての知見が既に入っていて分かるというのもあるかと思う。

Q：人間の頭の中には三次元のモデルを回す仕組み、見え方を変化させる仕組みが頭の中にあるからではないか。これを学習できれば、それを使って構成できるではないか？

A：それはどちらかと言えば、生成モデルである。

Q：CG を使った研究では、そのようなアプローチを考えたか？

A：三次元のモデルを作ってあるので、普通に CG で出している。しかしながら、いろいろ物の見た目や（できれば）形をきちんと学習した上で、違う視点を生成するという研究は十分考えられるし、敵対的生成ネットワーク（Generative Adversarial Networks: GAN）と言われるアプローチでも、そういったことはできると思う。ただ、それが学習サンプルとして本当に役に立つかどうかは、まだ微妙なところだと思う。

Q：深層学習で 3D 的な扱いをするテクノロジーとしては、いろんな見え方をした二次元サンプルを使って強引にやるというのは分かる。しかしながら、三次元的な要素を加味したような深層学習のようなものはあるか？

A：三次元情報を対象とした畳み込みニューラルネットワーク（CNN）は当然あり、距離画像を撮像した上で、椅子や机を認識している。

2.5 機械学習型システム開発のビジネス的課題

佐藤 洋行（株式会社ブレインパッド）

まず、自己紹介をしたい。私は農学部の出身で、その後、株式会社ブレインパッドというデータ活用に特化したベンチャー企業に入社した。そこでデータ分析や、今回のテーマである機械学習を組み込んだシステムを構築する際のプロジェクトマネジメント等に従事し、自社開発によるデータ活用サービスの開発に携わっている。また、最近では多摩大学にも在職しながら業務に取り組んでいる。

一貫して工学の門外漢であるが、ブレインパッドで 10 年近く機械学習周りのビジネスに取り組んできたので、そこで感じている現場としての機械学習型システムに対する課題というものをビジネス的な視点（営業や企画というような視点）から、お話ししたい。

このブレインパッドという会社が何をやっている会社かというと、まさに機械学習でシステムを構築して提供するというイメージを持っていただければと思う。

それを実現するために、三つの領域の実績をそろえている（図 2.5.1）。ビジネスとアナリティクスとエンジニアリングであり、データサイエンティストと呼ばれる人材は 70 名程度で、エンジニアリングという点では DMP というマーケティング領域の、あるサービスの中では国内ナンバーワンシェアのものを提供できる程度の技術力を持つようになった。

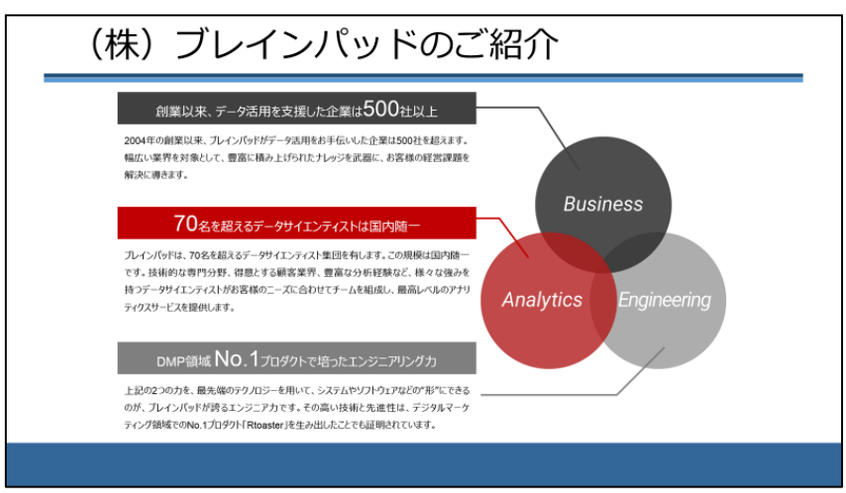


図 2.5.1　(株)ブレインパッドの紹介

私が従事するのは、デジタルマーケティングと呼ばれる領域で、ウェブサイトやウェブ広告などを駆使したマーケティングを支援するサービス。自社開発の汎用的なサービス、プラス関連したシステム構築やコンサルティング、あるいは汎用的なサービスから少し踏み出したような個別の分析などをトータルに提供している。その立場から見て、「機械学習応用システム」と呼ばせていただくが、その開発にどのようなビジネス的な課題があるのか、ということが本日の話題である。

機械学習応用システムということをビジネス的な観点で話すにあたり、４つに分類した（図 2.5.2）。まず、従来型のシステムとの関係性で、非機械学習システムでやっていたものを置き換える、あるいはそれに追加する場合と、今まで人間がやっていたことを新たにシ

ステム化する二つの場合で、難易度は大きく異なると考える。また、谷さんのプレゼンでもあったが、人のアクションを支援するシステムか、あるいはシステムがアクションまで全部担うシステムかという、この４分類とすると、ビジネス的な難易度は、縦軸では新規開発の方が難しく、横軸ではシステムがアクションを担当する方が難しいと考えた。

それぞれの例については、「非機械学習システムの置き換え」で「人のアクションを支援する」ものとして、不正検知のアラートが考えられる。「非機械学習システムの置き換え」で「システムがアクションまで担当」では、デジタルマーケティングにおける EC サイトでの商品推奨。従来は売れ筋ランキングのような情報提供だが、それを機械学習で置きかえるものである。「新規開発」で「人のアクションを支援」では医療画像診断、「新規開発」で「システムがアクションを担当」では、自動車の自律走行システムということになる。

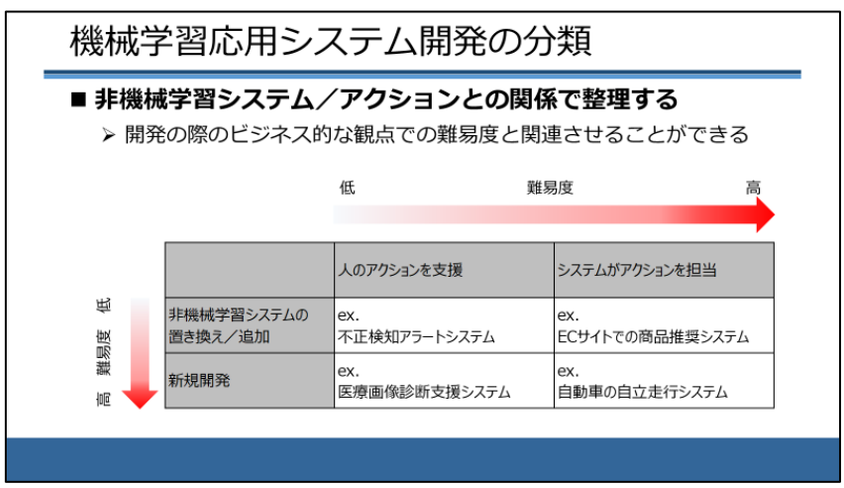


図 2.5.2　機械学習応用システム開発の分類

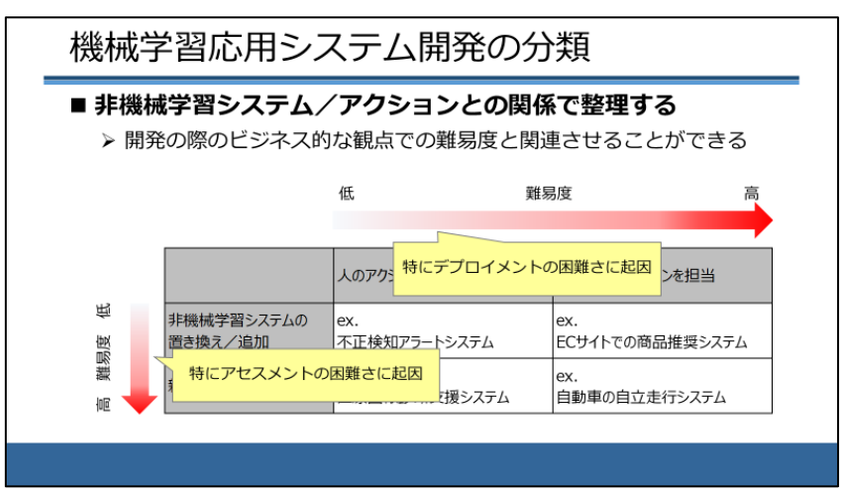


図 2.5.3　難易度の差の分析

このような４象限で分けたときに、難易度の差はそれぞれどこで出てくるか（図 2.5.3）。ビジネス観点で話しをすると、まず、非機械学習システムの置きかえか、新規開発かの違いは何かというと、アセスメントの困難さが異なる（図 2.5.4）。谷さんのプレゼンの中でも出てきた、どういうシステムにするか構想する部分での難しさというのがある。人がア

クションをするのか、システムがアクションまで担当するのかでは、デプロイメントで困難さが異なると思う。

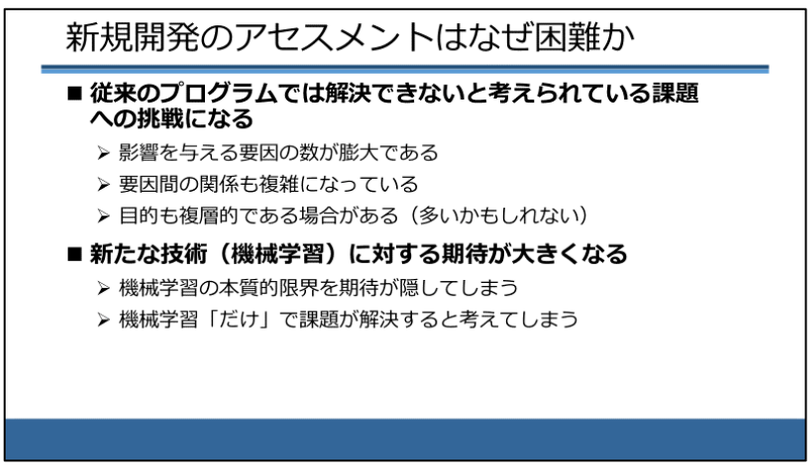


図 2.5.4　新規開発のアセスメントはなぜ困難か

まず、何で従来のシステムを置きかえるのに比べて、新規に機械学習型システムを開発する方が難しいのか。そもそも機械学習に限らず、何かシステムでできそうだと思うことは、大概すでにシステムになっている。では、新たに機械学習でシステムを作るのがどういうケースかというと、従来のプログラムでは解決できないから、新たな技術である機械学習で解決したい、ということになる。

なぜ従来のプログラムで解決できなかったかというと、影響を与える要因の数が膨大であることや、要因間の関係も複雑になっていること、あるいは谷さんのプレゼンにもあったが、目的が複層的になっていることなどである。最終的には売り上げアップが目的だが、そこにたどり着くための様々な目標があり、そういう複層的な目標というものに対して、システムを新規で作っていくところが難しい。

また、もう一つは丸山さんから「AI と言わないほうがいい」というお話しもあったが、新たな技術、特に AI というと、一般的に期待がとても大きくなる。丸山さんもおっしゃっていたとおり、機械学習には本質的な限界というものがある。そういう限界というものを大きな期待が隠してしまうところがあり、機械学習だけで課題は解決するだろう、機械学習が全部よしなにやってくれるだろう、と思われる点が、難しいところである。

私が実際に直面した事例としてお話しをすると、回転ずしのネタの需要の予測をしたいという要求があった（図 2.5.5）。回転ずしのネタの廃棄率を低下させるため、すしネタの需要予測のシステムを作ることになり、準備されたデータは、ID つきの POS データ、誰がいつ、何を注文したかということがわかるデータと、各種マーケティングデータ。課題は、毎日の各ネタの注文数を精度高く予測することとなる。

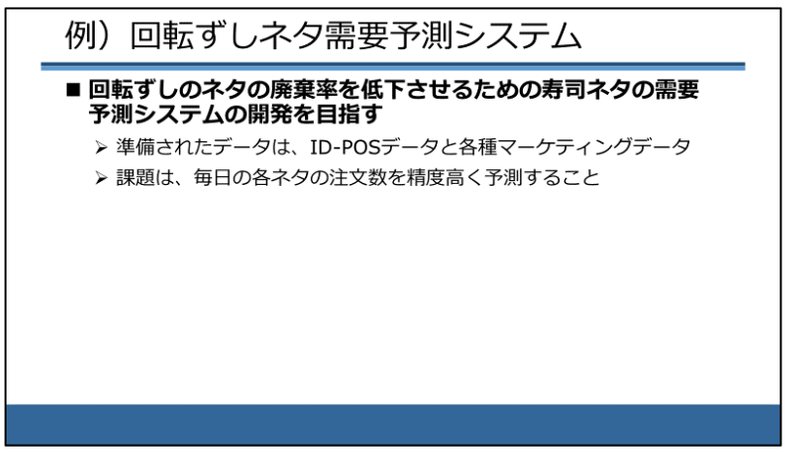


図 2.5.5　例）回転ずしネタ需要予測システム

このような要件の案件について、実現できそうだと思ってはまずい。実際にはかなり難しい。なぜ難しいかというと、さきほどの要因の膨大さと関係の複雑さということ（図 2.5.6）。ネタの注文数を決める要因に何があるかというと、膨大にある。そもそもどんなお客さんが相手になるのか、１店舗１店舗の商圏内に在住している潜在顧客の数、属性、嗜好性、あるいは来店動機など。マーケティングのデータがあるので、店側から仕掛けたものについては動機が分かるかもしれないが、例えば SNS など店側が仕掛けないものが動機になることもあり、これは挙げ出したら切りがない。

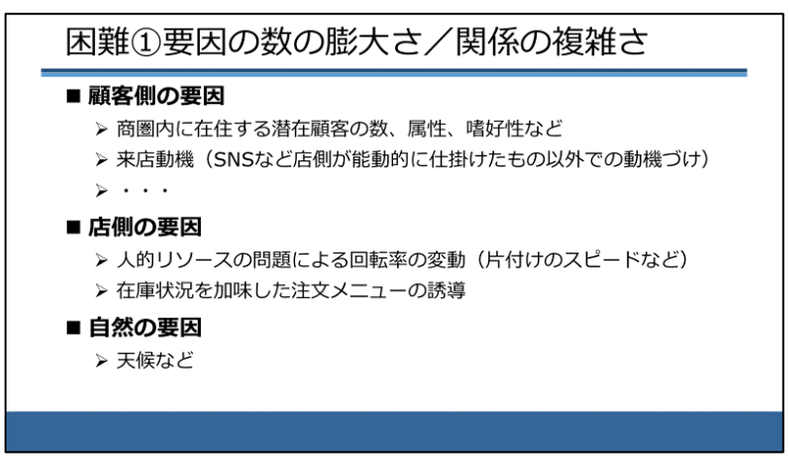


図 2.5.6　困難①要因の数の膨大さ／関係の複雑さ

また店側の要因もある。例えば、あるネタが売れていないときに、たまたまアルバイトが足りずに片づけが追いつかず、顧客をテーブルに案内できなかったということもあり得る。また、あるネタの在庫が少なくなってきたから、別のネタを多く出しておこうとか、ブロックの切り身が余りそうだから厚く切っておこうということもある。そのような店側の要因に加え、自然の要因もある。雨の日は来店が少ないなど、本当に多くの要因がある。

この多くの要因を全て取り入れるわけにはいかず、ID-POS データとマーケティングデータだけでやってくれ、というのが妥当なのかどうか。この判断も相当大変。どこまでのデータを収集すべきか、そして収集するコストに見合った成果を出せるのか、という点

を考えるのは相当に大変なこと。したがって、当初の要件というのは、全く要件になっていない。詰めておくべきことがたくさんあったということになる（図 2.5.7）。

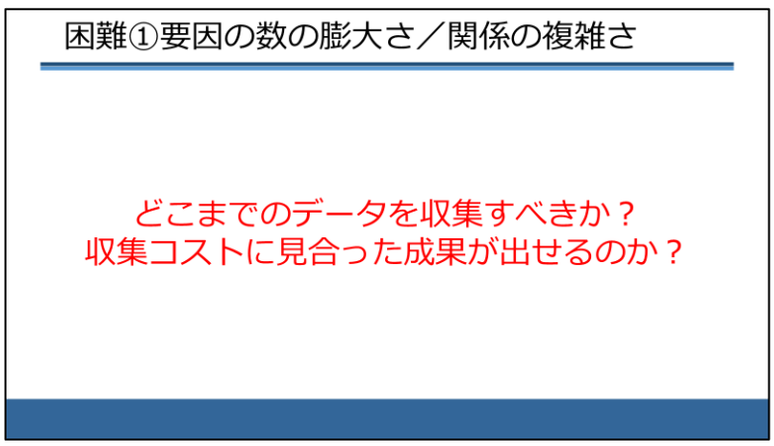


図 2.5.7　困難①要因の数の膨大さ／関係の複雑さ

もう一つ、機械学習の特性への理解という困難さがある（図 2.5.8）。先ほどのプレゼンにあったように、予測であればどうしても誤差は出る。その誤差をどう扱うか、ということも事前に決めておかないと、プロジェクトは失敗してしまう。

今回のネタの需要予測を考えるとき、ネタごとに予測誤差に対するリスクは異なる。例えば、マグロが品切れだと言われた場合と、エビが品切れだと言われた場合では、顧客の心証は全く異なる。恐らく、すし屋に行って、エビがないと言われるよりも、マグロがないと言われる方が顧客満足度は下がるだろう。すると、エビは過大評価に対してリスクが大きくなるが、マグロは過少評価に対してリスクが大きくなる。そういう点を最初から考慮しておかないとならない。

さらに、機械学習による予測をする場合には、いつ、どのくらい先の、どの数値を予測するか、ということも重要だ。それをきちんと設計しなければ、予測はできても実用はできない、ということになりかねない。この事例でいうと、例えば注文と仕込みの単位の違いということもある。だから、注文数を正確に予測することが、最終的な目標である廃棄率の低下に結びつくのかどうか、ということから考えなければならない。加えて、すしネタには毎日毎日、仕入れた日に使い切ることが前提のものもあれば、冷凍ものもある。冷凍ものの場合、仕込みの時期と注文の時期は時間差がある。すると、毎朝注文数を予測するシステムは意味がないかもしれない。それでは廃棄率が下げられないこともあり得る。

このように、予測のタイミングはいつがいいかという課題が出てくる。注文と仕入れは時間差があり、注文と仕込みにも時間差があり、それにどう対処したシステムを作っていくのか。機械学習はどの時点で、いつまでの予測をしなくてはいけないのか、ということを考えることが重要。こういう点を、機械学習を知らない人が考えようとすると難しい。

とても重要だと思うのは、今のところ機械学習の果たす役割は残念ながらきわめて限定的である。緻密な設計が必要となる。その点を理解しながらアセスメントを進めることが必要だが、期待ばかりが先行している状態であり難しい（図 2.5.9）。

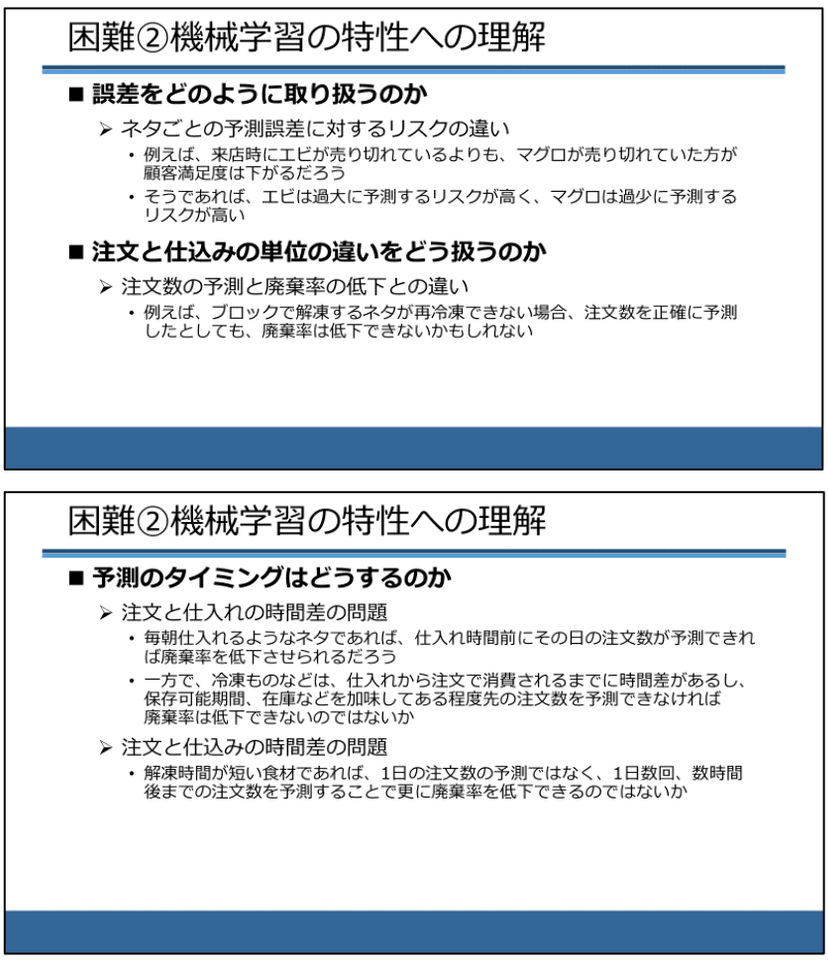


図 2.5.8　困難②機械学習の特性への理解

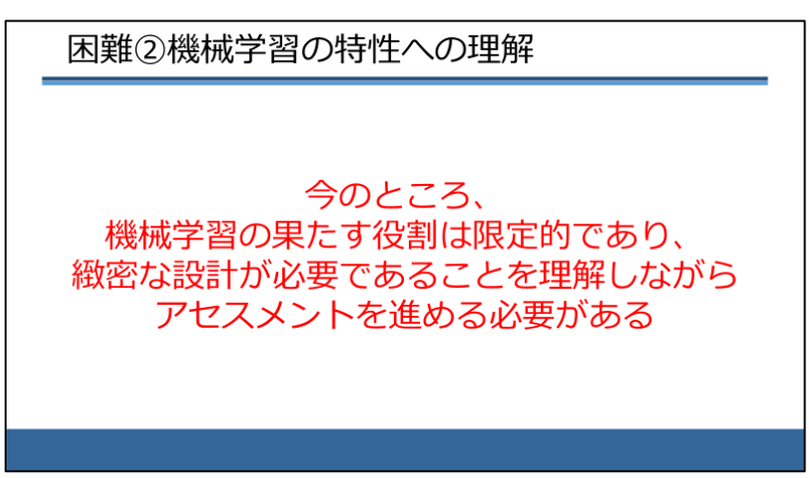


図 2.5.9　困難②機械学習の特性への理解

次は、人のアクションを支援するのか、システムがアクションまで担当するのか、というデプロイメントの困難さについてである（図 2.5.10）。このデプロイメントの困難さというのがどこで発生するかというと、この一言に集約されると思うのだが、従来のプログラムに比べて機械学習というのは直感的に理解しづらい。

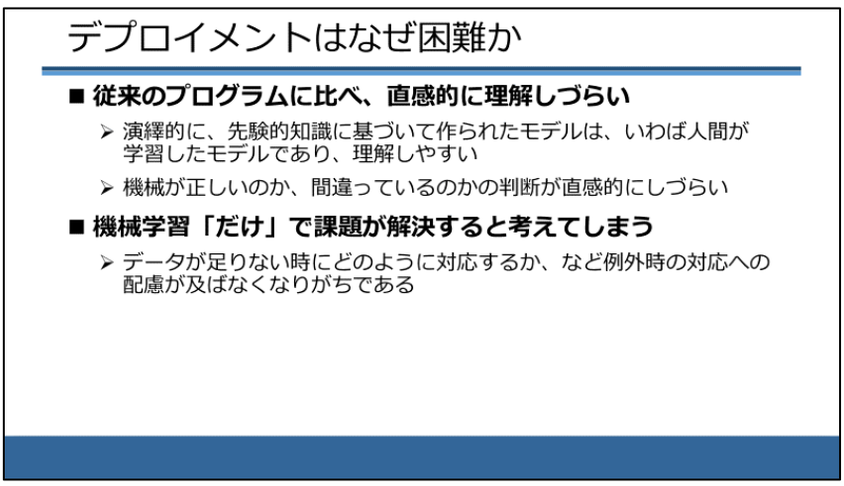


図 2.5.10　デプロイメントはなぜ困難か

例えば、データを分類することを考えたときに、単純に円で境界を描けるようなものはイメージしやすいが、実際に機械学習が扱うのは境界面がすごく複雑になる。そういうことをイメージするのは難しくて直感的に分からないため、アウトプットに対しても正しいかどうか直感的にわからないということがあり、人間とのコラボレーションが難しい。

もう一つあるのは、機械学習だけで課題が解決すると考えてしまう、という点。具体例でお話しすると、例えば EC サイトでの商品推奨である（図 2.5.11）。Amazon で行われているような関連商品などの推奨システムというものを考える。もちろん、その目的は明快で、EC サイトで売り上げを増加させたい、だから商品推奨システムを開発したいということになる。

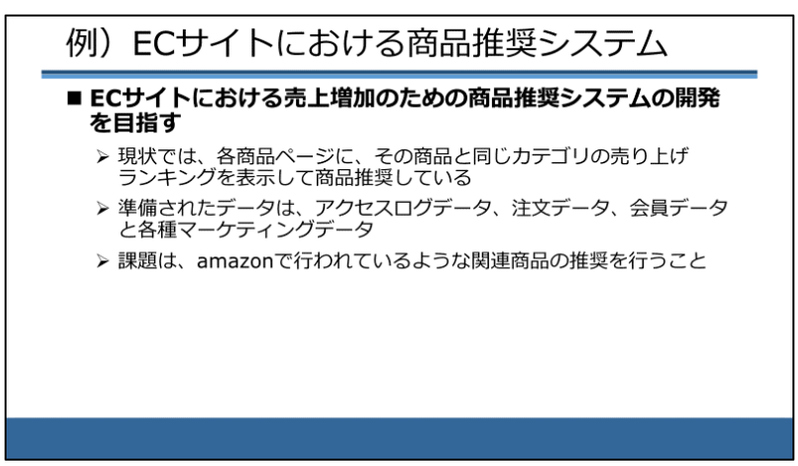


図 2.5.11　例）EC サイトにおける商品推奨システム

例えば現状では、各商品ページでその商品と同じカテゴリーの売り上げランキングを表示して、売れ筋の商品を推奨している。準備されたデータはアクセスログデータ、そのサイトでどういうページにアクセスしたかということと、何を注文したかという注文データ、会員データあるいは各種マーケティングデータがある。課題としては、Amazon がやっているような関連商品の推奨を行いたい、となる。

この案件も、一見簡単そうに思えるが、実際にはとても難しい。最初に出てくる問題は、どの商品が推奨されれば正解なのかがわからない点。そのカテゴリーのランキングを表示することはわかりやすい。しかし、機械学習で、人の行動ベースで関連商品を出すとなると、例えば服を扱うアパレル EC で、大人用カーディガンのページに関連商品として子ども用ズボンが出てきた。これが正しいか直感的には理解しがたい。では何らかの KPI を設定して正しいかどうかを測定しようとなるが、その KPI をどうするか、これも難しい。たとえ単純に、実際に推奨された商品が買われればいいとしても、何%一緒に買われればいいのか決めるのは難しい。

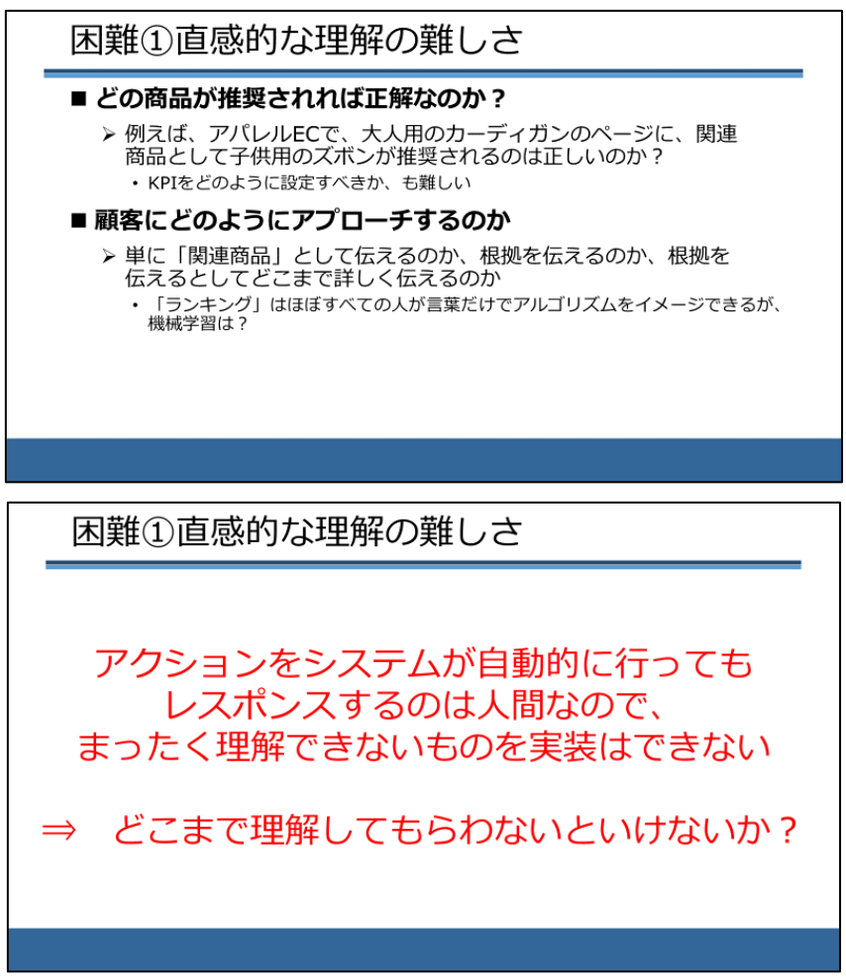


図 2.5.12　困難①直感的な理解の難しさ

次の問題は、この関連商品というわかりにくいものを、ビジネス側がお客さんに対してどうプレゼンテーションするのかという点。顧客から見たときに、推奨された商品が、自分の見ている商品、あるいは自分の興味に関連がなさそうだと思えたら、満足度は下がってしまう。それでも、「売れ筋ランキングです」と言われて推奨されれば、自分の興味に関係なくても仕方ない、と思ってもらえるだろう。つまり、機械学習による商品推奨をどのように顧客に説明してプレゼンテーションするか、というのもこの案件の成功には重要なのだ。決して良い機械学習のアルゴリズムが完成すれば成功、というものではない。

アクションをシステムが自動的に行うとしても、レスポンスするのは人間である。従って、理解できないというものは実装できない。丸山さんもおっしゃっていたが、どこまで理解してもらうのかという問題。いい説明は何なのかということは、本当に真剣に考えなくてはいけないのではないか（図 2.5.12）。

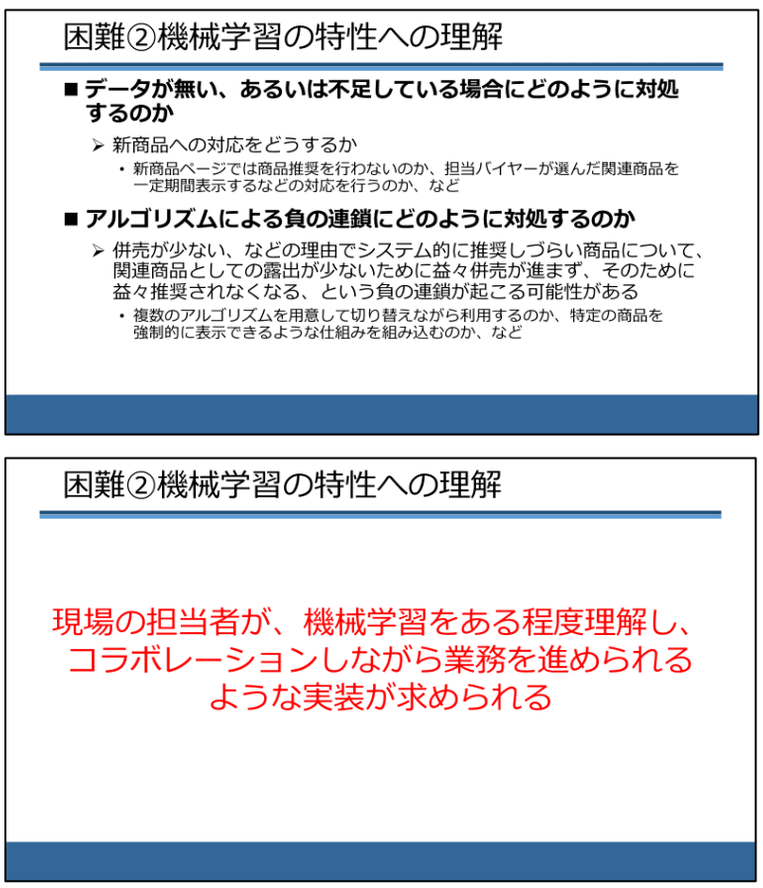


図 2.5.13　困難②機械学習の特性への理解

デプロイメントでの、機械学習の特性への理解ということに関連する困難さ（図 2.5.13）という点では、EC サイトの自動推奨で課題になるのが、新商品への対応である。データが無い、不足している、という場合に、どう対処するのかということ。こういう場合には、以前のランキングを出すのか、それとも、新商品に対しては人が推奨商品を決めて出すのか、どのように不測の事態に対応するのかということも、あらかじめ決めておかなくてはならない。

さらに、アルゴリズムがアクションまでやってしまうと、そのアクションによって学習データが歪んでくることが生じることもある。EC の推奨商品である商品を推奨すると、推奨された商品は売れやすくなっていく。一方、推奨されない商品は売れにくくなっていく。すると、売れにくくなった商品は併売もされなくなるため、次にもう一回、学習したときには、全く推奨されなくなる。このように、アルゴリズムがアクションした結果自体を学習に使っていくときに、何らかの負なり、正なりのフィードバックがあり得る。それにどうやって対処するのかということも考えていかなくてはいけない。

従って、専門家だけがわかっていればいいという世界ではない。現場の担当者に、ある程度、機械学習というものを理解してもらわないといけない。さらに、その人たちとコラボレーションしながら、業務を進めていけるようなシステムを作っていかなくてはいけない。その点が、機械学習応用システムの難しいところなのではないかと感じている。

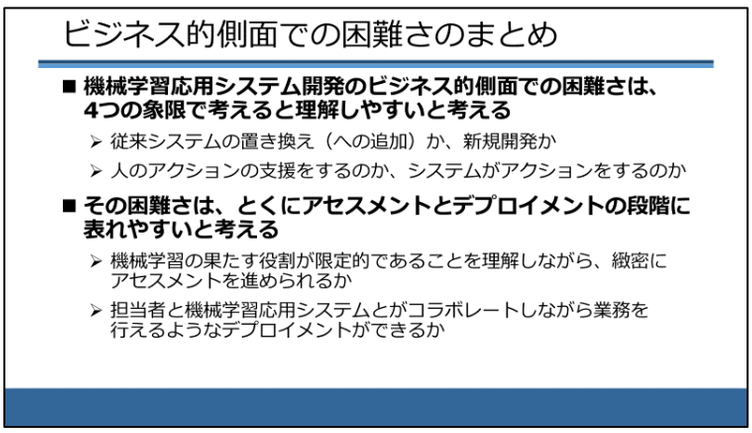


図 2.5.14　まとめ

最後に、機械学習応用システムについて、ビジネス的な側面での困難さという点からまとめてみた（図 2.5.14）。まず、４つの象限で考えると理解しやすいのではないかなということで、従来システムの置きかえあるいは追加なのか、新規開発かというところで異なること、人のアクションを支援するのか、システムがアクションするのかというところで異なること、とまとめた。

これらの困難さがなぜ異なるのかというと、アセスメントとデプロイメントの段階で顕在化しやすいのではないか。まず、従来型システムへの置きかえか、新規開発かというところで困難さがなぜ違うのかというと、後者では、機械学習の果たす役割が限定的であることを理解しながら、より緻密にアセスメントできるかどうかということが求められるので難しい。人のアクションを支援するのかシステムがアクションするかの違いというところでいくと、後者では、実務担当者と機械学習応用システムとがコラボレートしながら、業務を行えるようなデプロイメントというものをしなくてはいけないという点が難しい。

以上で、私の発表を終わらせていただく。

【質疑応答】

Q：アセスメントとはどういう意味で使われているか。

A：アセスメントというのは、顧客の持つ課題をどういうシステムで実現していけばいいのか、ということを考える段階。丸山さんがおっしゃったプロセスで書かれていた「アセスメント」、中島先生がおっしゃった「スカルセッション」、谷さんがおっしゃった「構想段階」と共通するイメージ。

Q：現場でシステムを使う人とのコラボレーションが必要で、その人が機械学習システムの特性を理解しなくてはいけない、という点は共感するが、佐藤さんのご経験で、現場の人たちはどうすればわかってもらえるのか。
A：いろいろパターンがあるので難しいが、一番効果的なのは啓蒙的な活動。自動推奨であれば、Amazon が大成功したという事例があるため、担当者も学ばなければならないというモチベーションが高い。なので、成功事例をどんどん作り、自分もやらなければまずいぞと思わせる、モチベーションづくりが一番重要ではないかと思う。

Q：システムの作り方において、透明性や、インターフェースの設計の仕方などの工夫はあるか。
A：例えば商品の自動推奨では、商品を自動推奨した結果、それが買われているのかどうか、その商品のページの閲覧数が増えているのかどうか、効果測定のインターフェースを用意している。それを担当者が見ることによって、自動推奨されるとページ閲覧が増えるんだな、買われるんだな、ということがわかるように工夫している。

Q：EC サイトではトレンドという要素もあると思うが、その対処や取り込み方はどう考えるか。
A：その意味でいうと、直近、いつまでのデータを学習データとして使用するのか、という点を考慮したりしている。直近のほうのデータを重視するのか、あるいは、ある程度長いスパンのデータを重視するのかということで重みづけをする、というのも一つ。他にも、明らかに季節性があるものなどについては、データセットを変えることで対処している。

Q：現場の方がそういったことも機械学習の側面からわかっていればうまくいくということか。
A：その通り。それをうまく引き出す力が必要となる。顧客からそういう情報をうまく引き出せるような人材が重要。

Q：今回のプレゼンでは、品質という言葉があまり出てこなかったが、アセスメントやデプロイメントのやりとりの中で、吸収されているような印象を持った。品質問題に対して、どう取り組んでいるのかという視点でコメントをいただきたい。
A：それは本当に頭を悩ませていて、言いたいことはいろいろあるのだが、なかなかまとめられずに省いてしまった。商品推奨という例では、プロセスでしか品質を定義できない。こういう論理で結果を出す、ということしかなくて、それで正しいかどうか、というのは定義できない。例えば、併売率がちゃんと上がっていればいいのかというと、実際の EC サイトはさまざまなプロモーションが並行しているので、その商品推奨だけで併売率が決まるということはあり得ない。
なので、先ほど述べたように KPI を決めことは相当に難しく、恐らく、成果報酬が難しいというところにもつながると思う。実際のビジネスは、何かシステムがやるといったときに、その他の要因で変動するものもあるので、商品推奨のようなものだと、プ

ロセスで品質を保証するとしか言いようがない。
予測ということでは、予測精度を担保するというようなことになる。もちろん、バリデーションというのもそうであり、一回システムを作って未来まで予測して、実際に起こっている未来で確認してもらうということしかない。

C：品質というのは、何を作るかというリクワイアメントそのものも品質だと思う。テスティングだけではない。冗談のような言葉で、バグがあってもそれはフィーチャーだ、と言うことがある。でも全く逆で、顧客に好まれないフィーチャーはバグ。その意味では、今回のお話はリクワイアメントのところ。しかも、非常に多くのステークホルダーがいて、丸山さんの言葉でいえば、帰納的にシステムをつくらなければいけない。そういうものに対する品質の問題だと思う。

Q：４象限で分けるやり方、あの説明は大変わかりやすかった。縦軸にいくほう、横軸にいくほうの難しさは説明されていたが、右下にいくときはどうか。

A：難しいこともあり、あえて今回は踏み込まなかった。自動車の自律走行システムについても考えるところがあって、これを自分が作るとしたらどうするのかというと、すごく難しいと思う。例えば、自動車の役目は何かと考えたときに、安全に目的地にたどり着くことだけではない。たくさんの要素がある。走っていて風景が気持ちいいとか、美味しいご飯が食べられるお店がたくさんある道を通りたいとか。
ナビを使っていてもそう。ナビはあまり満足のいく答えを出してくれない。本当に知らない道に行ったときは、ナビどおりに走ればいいという安心感はあるが、知っている道を走るときにナビはあまり役に立たない。なぜなら、例えば今、何千円を払ってでも 5 分早く着くルートが知りたいというような要求を柔軟に取り入れたルート提示をするようなことがない。そういったことを考えると、車がどう挙動すればいいのかを決めるだけでも結構大変だと思う。
その上にアクションを担当するということになっても、もちろん誤差があるわけで、しかも命にかかわることだから、何らか不測の事態に備えなくてはいけない、ということを考えなくてはならない。もちろん、そういう不測の事態というのを防ぐために、さまざまなものと関連性を持たなくてはいけない。例えば道路などにセンサーが設置され、それとのコラボレーションなど、その他さまざまなコラボレーションをやらなければならず、あまりに難しいと思って今回は省いた。

Q：ビジネスモデルの話で、むしろ作った後の継続的なアフターサービスというか、顧客を抱え込めるモデルではないかなと感じたが、どうか。

A：目指すべきところはその通り。私の部署は自社サービスということで、ある程度、汎用化したものを扱っている。一方でほかの部署ではデータサイエンティストが個別にデータ活用のためのコンサルティングや、個別のシステムを開発している。でも、そこは丸山さんや谷さんがおっしゃったように、POC 天国というか、POC だけで終わるということがたくさんある。必ずしもストックモデルだけにならないというところは、苦しいところなので、むしろ POC 地獄と言った方がいいですね。

3. 議論

3.1 未来社会創造事業の次年度公募テーマに向けた問題意識

前田 章（科学技術振興機構 研究開発改革推進部）

未来社会創造事業は平成 29 年度から始まった新しい事業である。本事業の特徴は基礎研究から社会実装、実用化までを見据えた研究を実施すること、研究開始時から実施内容を決め打ちで進めるのではなく、探索研究期間を設けた上で、スモールスタートで始めて本格研究に移行するという仕組みを導入していることである。

本事業には四つの領域があり、そのうち「超スマート社会の実現」領域を担当している。超スマート社会の間口は非常に広いため、言葉の定義として、Society4.0 が情報化社会だとすると、Society5.0 は何か、4.0 との差分は何かということを仮説としておいて進めることとした。一般からの意見公募も踏まえて、CPS（サイバー・フィジカル・システム）、サイバー空間における情報処理だけでなく、実世界のモノとの相互作用による価値創造に焦点を当てて、超スマート社会を考えることとした（図 3.1.1）。

図 3.1.1 「超スマート社会」、「Society5.0」とは何か？

例えば、電力や交通システムなどの社会インフラがどんどん IT 化されて高度化するときに何が起きるか、自動運転やロボットがあったときに IoT デバイスはどのようになるか、そういう社会をイメージしている。Society5.0 までジャンプしておらず、4.1 かもしれないが、少なくとも将来の目指すべき方向性として置いている。補足するとサイバー空間は実世界と一体であり、クラウドと IoT が別々にあって、それらを繋ぐのではなく、私は CPS で一体的に渾然と進むイメージを持っている。

図 3.1.2　未来社会創造事業における「超スマート社会」の考え方

仮説なので正解であるかは分からないが、これを前提で考えると必然的に分散になる。インテリジェンスがモノに埋め込まれるということであるので、全部がクラウド処理でクラウド同士の連携というのももちろんあるかもしれないが、もう少し垂直方向のエッジも含めたレイヤーでの分散になるかと思っている。そのため、分散における機能の最適配置だとか、動的再構成、どの処理をどこに置くのかといった、システム的なあり方も随分変わるのではないかという課題意識を持っており、特に今年度は異種システムの連携、SoS（System of Systems）といった言葉をキーワードとして進めてきた。

今年度はこのような考えで公募を実施したが、来年度以降の新しいテーマを考えたときに AI は無視できない。このインテリジェンス化は当然、AI 的な、機械学習的な技術が支えるわけで、そうすると、学習によって様々なシステムの機能がどんどん変化し賢くなる。このときにシステムアーキテクチャ的に何が起きるのか、という課題意識を持っている。それが図 3.1.3 の「3.システム・ソフトウェア構造の変革」であり、新しいアーキテクチャが必要ではないかと考えている。

本ワークショップでは、システムの開発手法ということにフォーカスされているかもしれないが、機能がどんどん変化する、また、新しい機能が生まれるというときに、システムのあり方自身、アーキテクチャとしてどうなのか、というところに課題意識がある。本年度からサービスプラットフォームというテーマで公募を行い、研究がスタートしているが、まだまだ技術的な要素が不足しているので、こういった観点、IT・AI 的な観点から来年度の新しい公募テーマを検討している。言い直すと、分散だとか、AI、機械学習によってソフトウェア・システムのあり方が変わるのではないかという問題意識を持っている。

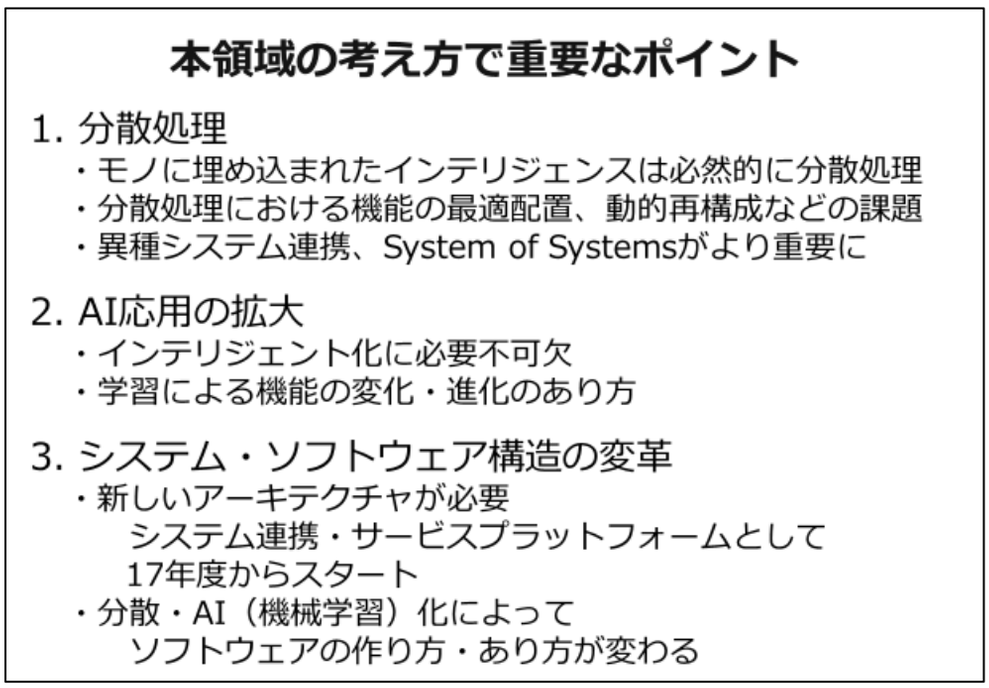


図 3.1.3 「超スマート社会の実現」領域の考え方で重要なポイント

私から見たときに関連する動きとして指摘しておきたいのがこの三つである（図 3.1.4）。

まず一つめ、私はオンライン学習に興味があって、もちろん、事前に大量データを一気に学習して性能のよいものを開発することになるが、それをデプロイした後の学習も必要になると思う。特に IoT では、無視できないと考えている。

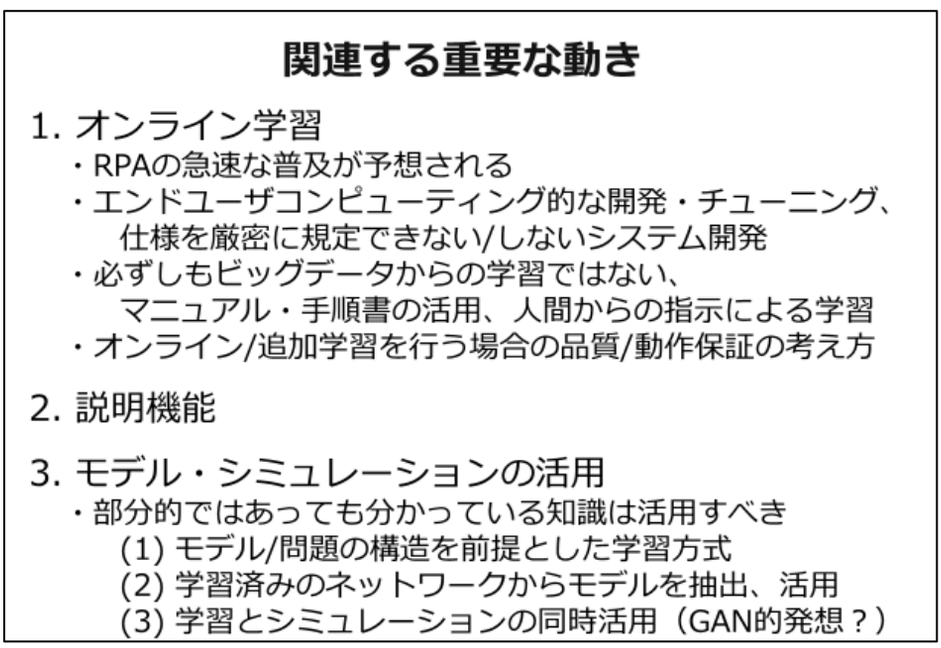


図 3.1.4 関連する重要な動き

それから、もう一つ、注目すべきだと思っているのは RPA（Robotics Process Automation）である。企業情報システムの中の特にフロントオフィス的なところの処理をどんどん自動化するというふうになってきているが、なぜまだできていないかというと、例外が多いだとか、仕様を書いてもコストに見合わないことが多いためだと思う。そのため、手作業で進めてしまう方が早い。しかし最近、メガバンクでものすごい合理化を進めるという話があるが、おそらく AI 開発的な手法を使うことでシステム開発のコストが下がるだろう。それを見込んで、RPA の普及というのがどのくらい Society5.0 に関係するか分からないが、これから大きな動きになるではと思っている。

以上のように考えると、エンドユーザコンピューティングのように、納めておしまいではなくて、納めた後も、昔のやり方でいうとシステム保守でお金をもらうと同時に、常にエンドユーザとつながりながらソフトウェア開発を回していくような仕組みになるのかもしれない。そこでは、当然、チューニング、追加学習、といった技術が必要になると思う。Society5.0、異種システム連携とどのぐらい関係するかはわからないが、こういう流れもあると思っているので、この周辺の技術にフォーカスした提案を募集できないかというふうに考えている。

これを考えると、必ずしもビッグデータだけではないと思っている。途中の議論でも何度か指摘させていただいたが、マニュアルや手順書が大事ではないか。昔、保険システムの開発に関して約款ベース開発技術を検討した事例を知っている。保険の約款からシステムの仕様の一部を自動生成できないかというアイデアである。もっと単純に考えると、例えば旅費精算の自動化をしたときに、これは間違っているよと人間が教えてあげるなど。そんなことを考えると今の学習とは違うやり方の仕組みが、これは既存のやり方でもできるかもしれないが、ビッグデータの初期学習に加えて、新しいアイデアで何かこれをやるとシステムの開発手順はどうなるのかというのが、このワークショップの議論にマッチするかと思って資料に書かせていただいた。当然ながら、これは皆さんの発表の通り、システムの品質や動作保証に関して、特にベンダーの立場からいうと、どこまでが企業の責任かという切り分けのやり方が随分変わってくるのではないかと思っている。

説明機能は、皆さんよくわかっていらっしゃるので省いたが、重要なポイントだと思う。

もう一つ、指摘したいのはモデルとシミュレーションで、部分的ではあっても分かっている知識は当然使うべきで、色々なやり方がある。まず、わかっていることを予め初期値として、深層学習（ディープラーニング）でもネットに埋め込むという方法や、逆に、学習済みのデータから、問題の構造を抽出する仕組みというのはあるかもしれない。それから、学習とシミュレーションについて、GAN（Generative Adversarial Networks）という話があったが、データの生成のほうにモデルとシミュレーションを使って、同時にモデル・シミュレーションのパラメーターのチューニングも行うという発想もあるかもしれない。

しかし、ビッグデータだけからやるという発想は、多くの人がそちらへ向いているとすると、モデルだとか人間が持っている知識だとか、構造みたいなものを使うという発想がないかという問題意識を私自身は持っている。この問題意識を超スマート社会という文脈の中で、次の技術開発の課題としてどのように定義したらよいかというのを、今、議論しているので、ぜひ、皆様方の今日の議論を踏まえてフィードバックをいただければと思っている。

【質疑応答】

C：本領域の考え方（図 3.1.2）について、CPS をどう捉えるかは人によってさまざまだが、私もプレゼンの中で申し上げた通り、個人的な思いとして、CPS はソフトウェアのパラダイムだと思う。オブジェクト指向というのが 20 世紀の終わりから始まって、随分、流行し一般化された。オブジェクト指向のオブジェクトは結局、還元主義である。粒々に分けていくことで、それが様々なところに使えるような汎用な部品になる。それで、再利用性という言葉が出て、ソフトウェア開発の効率向上に寄与した。

そのときに CMU（Carnegie Mellon University）のメアリー・ショー（Mary Shaw）たちが、オブジェクト指向にそぐわない、マッチしないソフトウェアもたくさんある、ということを議論していた。つまり、オブジェクト指向の考え方ではうまくシステム開発ができない場合があることを CMU の SEI（Software Engineering Institute）の研究者が随分議論していた。そういう流れから考えると、CPS というのはシステムパラダイムである。Cyber Physical System は、NSF のヘレン・ジル（Helen Gill）が言ったことで、彼女は NSF の前に DARPA で SEC（Software-Enabled Control）というプログラムを担当していたが、これはまさに制御工学における制御方法のコンピュータライゼーションである。モダンな分散コンピューティングのテクノロジーを使って、制御をいかに進歩させられるか、という研究を実施していた。

そういう流れの中で、なぜ Cyber Physical System、Cyber とつけたかというと、UC バークレー（University of California, Berkeley）のエドワード・リー（Edward A. Lee）の本によると、その根源はサイバネティクスと同じ発想だということである。ノーバート・ウィナー（Norbert Wiener）のサイバネティクスは人工系（機械系）、自然系を含めて共通的な物の見方があり、それは実はコミュニケーションとコントロールである。そのコミュニケーションとコントロールが SEC の研究の後にもう一度、出てきた。それによって新しいシステムをつくるパラダイムとして CPS を考える。多分、このような理解が最も抽象的で、私にはしっくりくる。

そのように考えていくと、本領域の考え方というのは割と私にもしっくりきている。その中でいうと、現実世界の実世界の現れとしてのデータ、これをいかに取り込むかがもう一つの CPS のポイントである。機械学習的なアプローチ、データを中心に物事を考え、そこからいかに有用な情報を取り出すか。これは CPS の重要な要素技術だと思う。

A：センシングが重要というのはよく分かる。今日、何人かの方がおっしゃったとおり、ビッグデータの専門家はどのデータをとるべきか、とったデータはどういうコンテクストをとったものか、など、そういうところに踏み込まないと、おそらくまともなデータはできない。フィジカルな世界とサイバーの世界がうまくマッチするような仕組みが必要になったということで、モデル・シミュレーションも同じ問題意識であると思うので、このあたりの技術も含めて何か一つの公募のテーマにできればよいと今は考えている。

3. 2 総合討論

福島：趣旨説明に用いた資料の最後のパートに、これまでに実施した有識者十数名のヒアリングで得た主なコメントを列挙しておいた（図 3.2.1）。1 時間以上かけて詳しく話を聞いたケース、立ち話的なケースもあり、ここでは有識者名は省略させていただく。

1. AI の社会実装（＝事業化）においてリスク最小化のマネージメントが重要。そのため機械学習型システム（PL 法やリコール制度の対象になり得る）の品質管理基準（通常有すべき安全性とは？）が必要。
2. 工業的規格がないと取引が困難。学習プロセスを受託する事業を想定した場合、学習済みモデルに対する品質基準（受入基準）をどうするか？
3. 機械学習型システムや学習済みモデルの競争力の源泉としても、品質基準が必要（粗悪品に対する明示的な差異化を可能にする）。
4. 機械学習型システムは、要件の厳密な記述ができない、モジュール毎の検査ができない、学習用データが実データの統計的に適切なサンプルであることが保証されない。
5. 産業界での問題意識は高まっていて、非常に重要な研究だが、研究者はまだ少ない。
6. システム開発方法論としての研究はまだ少ないが、深層学習の解釈性問題、騙す攻撃、敵対的生成ネットワーク（GAN）に関する研究等、方法論や品質テスト･限界テストにつながる研究は活発になっている。
7. 機械学習型システム開発を推進するには、精度･品質等のための研究開発だけでなく、データを集めてアノテーションするという泥臭い作業のための方法論やツール開発も重要。
8. 深層学習（帰納）＋記号推論（演繹）の統合アーキテクチャは、AI 分野の重要な研究テーマであり、システム開発方法論につながる基礎研究と考えられるかもしれない。
9. 機械学習型システムの比率はまだまだ少ない。従来型システムの大規模複雑化で生じている品質問題の方が重要。
10. システム全体が Bigdata-defined になることはほとんどなく、メインは Software-defined であり、それらの組み合わせとトップダウン設計を中心に置いて考えるべき。
11. 使われながら機械学習して動作が変わるのは怖いので（MS のチャットボット Tay の事件）、それを許可する顧客はいない。事前に用意したパターンのランキングを変える程度。
12. 命に関わる部分と利便性に関わる部分を分離。機械学習型モジュールは 100%保証ができないので、同じ機能に関する別アプローチや問題発生時のリカバリー策も併用している。
13. 100%保証は無理でも、納得性･受容性は確保し得るはず。
14. システムとして組み上げる上で、現象モデル、計算モデル、アルゴリズム等を区別し、何の問題なのかを整理していくべき。

図 3.2.1　有識者ヒアリングでの主なコメント

本日の発表の中でカバーされていたものもあるが、カバーされていないもので重要な指摘も含まれている。これから行う総合討論では、これらのコメントも取り上げ

ていきたい。
総合討論で議論したいポイントは、本日冒頭の趣旨説明の中で示した課題の全体像的なものについての改良アドバイス、いま述べた有識者コメントについての意見、それから、先ほど説明のあった未来社会創造事業を含めてこの研究課題に対する取り組みの進め方、という 3 点を考えている。

【品質問題について】

福島：まず、有識者コメントの 1・2・3 にもある課題。つまり、社会実装におけるリスク最小化が重要で、機械学習の社会実装は PL 法やリコール制度等にも関わる品質基準面の問題をどう考えていくか。ある種のシステムは、こういう点をきちんと考えておかないと、ビジネスとして回らないし、品質基準を定めておくことが粗悪なところの差異化、競争力の源泉にもなる。このあたりについて意見や補足があればお願いしたい。

丸山：品質に関して言えば、今の品質保証のやり方はプロセス品質である。そもそも、プロセス品質しか測れないような世界で、そういうフィールドが培われてきたからだ。でも、僕らは今、プロダクト品質をきちんと測れる好機にあり、その品質の測り方を根本から変えたらよいと思う。

福島：それはデータが品質のベースになるという意味合いか？

丸山：第三者機関がもし品質の評価をするのであれば、例えば日本の自動車アセスメント（Japan New Car Assessment Program: JNCAP）として実施されている自動車の安全性能の客観的評価のように、品質の客観的な評価が可能となる。今の自動車は、１台１台を全部ぶつけて検査しているわけではない。いろいろな検査をするにしても、抜き取り検査しかできない。それに対して、少なくともソフトウェアに関しては全部検査できるというところは非常に大きな違いであり、好機だと思う。

前田：機械学習かどうかにかかわらず、例えば認識のようなソフトウェアモジュールは過去から数多くある。監視でも故障診断でも何でもそうだが、品質の問題は（例えば、精度が 98%というように）当然クリアしていると思ってよい。このように、そもそも 100%の精度は期待できない世界があって、それに関しては従来から品質保証がされているとすると、画像認識や手書き文字認識など、機械学習が入ったことにより、何が変わるのか？

福島：従来のものでも、ある種のベンチマークに対しては、精度を客観的に言うことができるが、認識システムは、適用される現場によって光の条件等を含めて条件が大きく変わるので、さまざまな顧客の環境で必ずその精度が得られる保証はない。

前田：郵便番号の自動読み取りシステムがそうだが、それを設計するために、力まかせにアルゴリズムを書いてやってきて、今、それが動いている。そのような従来の世界と、ビッグデータを活用して機械学習でやろうとする世界には、本質的な差はあるのか？

中島：郵便番号読み取り等のある種ややこしい問題から離れて、もっと簡単に考えればいいと思う。普通にプログラムを書くときは、例えば、入力変数Xは 1 より大きくて 10 より小さいというようなプリコンディション（precondition: 事前条件）を書く。

プログラムを開発する側はプリコンディションに合うデータがやってくると思って処理を書くので、使う側はその関数を呼び出すときにはプリコンディションが満たされていることを保証する。これは、ある意味コントラクト（contract: 契約）だと言える。

機械学習の場合、機械学習のプログラムに突っ込むのはデータセットである。そのデータセットのスペックというか、形、それを今のディスクリートなロジックで書いたようなプリコンディションのような形で書けるのか、そこが一番の大きな違いである。それから、例えば、そのデータセットの分布が非常に綺麗な確率分布をしている場合、それに対して尤度関数（likelihood function）を計算してやれば、やってきたデータがどのくらい外れているかどうかが分かる。でも、今、機械学習が扱っているデータセットというのは、そのような綺麗な確率分布関数で書けるものではない。そこが本質的な違いだと思う。

結局、何を気にしているかというと、プログラムを実行したときにデータがやってきて、プログラムが異常終了したときにシステムによってコアを吐くかもしれないことである。コアを吐くか、吐かないかというのは、実はプログラムを開発するときには、プリコンディションを明確にすることによって入力のチェックができるので、違反する例外的なデータがやってきたら、プログラムを例外処理すればよい。それができるかどうかである。

前田：やはり、郵便番号認識の例が分かりやすいと思う。文字認識のアルゴリズムを書く場合、例えば縦方向と横方向でヒストグラムをとって重心を求めるようなことをする。そうやって作ったアルゴリズムのプログラムの品質保証をするということと、ご指摘のビッグデータで品質保証するということの差が私には分からない。もし同じだとすると、機械学習だからという論点はどこにあるのか？

中島：つまり、それは何が違うかというと、品質保証をするときに入力するテストデータが明確に厳密に定義できるかどうかである。データセットのサンプリングの話は機械学習の話ではなくて、統計的な一般的手法のことである。例えば、計量経済の分野でも、標本サンプリングのバイアスの問題はある。結局、今までディスクリートな世界、要するに離散数学の世界で物事を考えていたソフトウェアの開発者が、そのような統計的なリテラシーが必要不可欠な分野のものを作らなければいけない場合、おそらく今までのプログラムの開発手法そのものが非力になってしまうのではないか。

茂木：いろいろな自動運転システムは世界のあちこちで開発されている。それは、もちろん地域性はあるにしても、車はどこを走るかわからないので、どこにでも売れるようにするには、それを全部でばらばらでやっているのはもったいない。もし標準的な１台があったとすると、先ほど丸山さんがおっしゃった第三者による検定のための一種の評価用の入力セットになり、品質評価の基準の助けにならないかと思う。

中島：それは結局、特定アプリケーションドメインのベンチマークであり、ベンチマークデータセットをきちんと整理して、それを正式に認めるということである。例えば、車の自動運転のような非常に社会的な影響が大きいものであれば、それを作るための価値もあるし、コストをかけてもよいだろう。ところが、機械学習は自動運転だ

けのものではない。様々なアプリケーション領域に対して、同じように世界的なベンチマークになり得るようなデータセットを作れるかという問題だと思う。

福島：ベンチマークデータを作ることで、画像のベンチマークのように、アルゴリズムの優劣のようなものは一つの参考として出てくる。ただ、そこで良いアルゴリズムがあったとしても、それが適用される現場でその性能が出るかというと環境によっていろいろ変化するものだ。郵便の話に戻ると、私自身も以前に郵便読み取りシステムの開発をしていたが、現場では大量のデータが実際に流されるので、それで平均的であれ、非常に例外的であれ、起きた現象が全部分かるので、現場の人は満足しているのだと思う。つまり、郵便読み取りシステムは機械学習型だが、その性能評価が、毎日流れる、あるいは、過去に流れた大量データで確認・検証できるので、現場は納得できるのだと思う。

前田：それも品質保証の文脈でいうと、精度が 100%にならないのはお客様も開発者も分かっていて、ベンチマークデータにより精度が 99.9%であるという品質で納品することである。それを是とするのならば、機械学習でやるということで新しく生じる問題は何か？同じことを繰り返して言っているが、私には違いがわからない。

丸山：そういう意味では、機械学習の問題は新しい問題ではなくて、昔からある問題である。郵便番号の読み取りと同じタイプの問題であることをお客様に理解していただければ、それでいいのではないかと思う。

前田：今では適用するドメインもどんどん広がっているし、お客様の要求する基準も変わっているので、新しい品質の考え方が必要になるであろうという考え方でよろしいか？

丸山：古い品質のモデルを持ち込んでもいいと思うが、それをお客様が納得してくださるかどうかという問題である。

福島：外れたケースがたまに出てきた場合、顧客がたまには外れるものだと思って納得してくださっていれば大丈夫である。間違う例は出てくるので、そのときに、これを直して完璧にせよ、あれも直して完璧にせよとなると、これだけ直す、あれだけ直すというのが難しいという機械学習の品質問題になってくる。

【要件記述について】

丸山：有識者コメントの 4 番には異論があって、もちろん、生成系のモデルというのは評価が非常に難しいというのはわかるが、最適化系のものに関しては、実は要件の厳密な定義がないと、つまり効用関数が厳密に定義されないと、強化学習はできない。先ほどぶつからない車の話の際に動かない車が出てくると言ったが、そういうところが実は今までのシステム開発よりも、より厳密な要件が必要になってくると思う。そうでないと、今までの要件の隙間は、実のところ実装する人たちがある意味、常識で埋めていたところがあり、良くない点である。したがって、要件の厳密な定義が必要な場合もあるし、そうでない場合もある。

福島：その要件の定義の仕方は、従来型とはだいぶ異なる仕方か、それとも、要件の定義の仕方自体は同じようなものか？　このコメントは、要件の定義が違うということ

は開発の方法論が違うという文脈で出てきたように記憶しているので、要件定義はきちんとするが、従来のソフトウェア開発とは違うという理解になるか？

中島：たぶん、要件と言っているものの中に様々な要素が入っていると思う。つまり、機能的なものとか、効用関数のような概念もあるだろうし、どのようなデータセットを使うかという、データを使う側がデータに求める性質のようなものも、暗黙のうちに入っているのではないか。

福島：つまり、要件や品質をきちんと定義していかなければならない。

【品質管理とリスク管理について】

茂木：ぶつからない車の場合は、出力がスピードとブレーキなので、出力はゼロではいけないということか？

丸山：はい。ぶつかったときのペナルティが無限大ではダメで、トレードオフを明示的に指定しないと効用関数は動かないということである。

茂木：要件が厳密ではなくて足りない場合、人間であればプログラマーには分かるが、機械学習は単にプログラムなので、ある種そのような常識的なものが継承されない。

前田：その例では、衝突防止に関する厳密な定義ができるのか？

丸山：それはチャレンジングな点で、先ほどのロボットが人を殺してしまう例に関しては、実は効用関数を全部書き下すことは、いわゆるフレーム問題に相当するので、おそらくできないのではないかという気がする。システムの適用範囲を明確にすることは、クローズドな世界では、ある程度上手くいく。例えば、アルファ碁のような世界では上手くいくと思うが、オープンな世界では多分できないだろう。少なくとも今は解が知られていない領域の問題で、基礎研究のレベルとしてチャレンジすべき課題かもしれない。

藤吉：特に強化学習の場合は、報酬の設計というところが、たぶん人間の常識を与えるところになるので、それをしっかりしないと、先ほど言ったようなことが起きてしまう。だから、その中で探索すると（社会としては認められない）全然違うところにいってしまう場合があるので、そこを上手く設計することが重要になってくる。

佐藤：そういう意味では、中島先生がおっしゃっていたように、ソフトウェアの使い方を限定せずにより緩くするような世界で起こる問題である。機械学習をより限定的に使おうとしているところに関していえば、十分に要件は定義できる。先ほどの商品自動推奨は、まさにその通りで、人の併売行動によって関連した商品を出すという要件は定義できる。したがって、前田さんがおっしゃっていたように、少なくとも機械学習が出てきて問題になりそうな点は、ドメインが広がっているということである。

そのドメインによって多分、品質というものも全然違ってくるし、要件というのも全然違ってくる。そのような中で、システムの統一的な、あるいは抽象的で網羅的な学問みたいなものを作ろうというのが、多分すごくチャレンジングなところではないか。その突破口を開こうと思ったら、ある程度、分けて考えなくてはいけない。品質といってもどこの品質を指すのか、要件といってもどこの要件を指すのか、まず初めに定義をすることが必要である。

丸山：典型的なものをパターン化するようなアプローチはあるか？

佐藤：機械学習の使い道は結局、予測、判別（画像認識も含む）、マッチング、最適化の四つであると思っている。これらの四つに対して、それぞれ定義しなければいけない要件は違うし、求められる品質の確かめ方も違ってくる。画像認識というと、絶対にがっちり決まった正解があるという世界なので、いわゆる精度の話になると思う。しかし、予測では、いつまで何%の精度を保たなければいけないのかという時間の概念が入ってくる。例えば、１カ月、１年後まではこのアルゴリズムで何%を保証するが、その後、世の中の移り変わりにつれて、精度はどんどん落ちていくような話が出てくる。マッチングになってくると、何が正解か分からなくなってしまうというようなことが起こりうる。そうなれば、全く、品質というものの考え方が変わってくる。

中島：今までの議論は機械学習側から見たものであるが、改めてアプリケーション側から見たときには、例えば自動車だったら非常に厳しい品質が求められることになる。しかしながら、例えばマーケティング的なところで、お客さんにレコメンデーション的に示唆するようなものであれば、もう少し緩いものでもよいのではないか。それを示した表がレベル別に書けるような気がする。

例えば、情報技術セキュリティのコモンクライテリア（Common Criteria）では、EAL（Evaluation Assurance Level）と呼ばれる保証レベルで１から７までランク分けがされており、システムが適切に設計され正しく実装されていることを評価する尺度を考える枠組みである。機械学習の問題に関しても、同じような形でレベル分けをして、そのレベルが満たされているかどうかをチェックする第三者の評価機関が必要である。

また、機能安全の話もそうだが、例えば自動車でいうと ISO 26262（自動車機能安全規格）も幾つかのレベルに分けられている。社会実装されるものに対して 100%の品質保証は無理なので、いわゆるディペンダビリティの観点から、発生頻度と影響力の掛け算で物事をきっちり整理して、それに見合ったコストをかけるような形で物事をまとめていかなければならない。

セキュリティであれば、非常に長い時間（おそらく 10 年、20 年の時間）をかけて、コモンクライテリアとして EAL1～EAL7 の 7 レベルにまとまっている。残念なことに、機械学習の話は、今始まろうとしているものだから、まだ見えていない。コモンなどあり得ないからこそ、今やるべきである。

佐藤：その話はすごくグサッときた。品質管理とリスク管理は、ソフトウェア工学の中でどのような関係性にあるか？

中島：非常に素朴に言うと、考えなければいけないディペンダビリティの性質としては信頼性と安全性があるが、品質管理の中心は信頼性に関連している。つまり、何を作るかが決まったときに、決められたとおりにシステムが動くことだ。一方、（エンジニアリングとしての）リスク管理の場合は少し違っていて、それの範囲を超えたときにも周りにダメージを与えないことであり、どちらかというと安全性に関連している。

佐藤：そこで、先ほどいろいろ話されていた品質管理という中に、リスク管理が入っているように聞こえたので、そこは議論として分けるべきではないか。先ほどの有識者コメント 1 ～4 番の中でも品質というように言っていたが、その品質の中にリスクが入っており、リスクと品質が混在していると思われる。

中島：先ほど、要求のところも重要で品質問題であると言ったが、一般的には、信頼性と安全性を区別せずに語ることが多い。しかし、これは実は安全性の問題ということもある。なぜならば、期待されないシステムは品質が悪いとも言えるからだ。期待されないということの中には、安全性を壊すということも入る。逆に、別の言い方をすると、システムがもしマルファンクション（malfunction: 機能不良・不調）したときに、外部に被害を及ぼさないためには何をすべきか？というのは安全性の問題だが、これは要求仕様のときに決めなければいけない。

福島：それに関連して、有識者コメントの 12 番では、自動車にも機械学習が入ってきているが、アーキテクチャ的に命にかかわる部分と利便性にかかわる部分は分けているとか、機械学習は 100%保証ができないというのはよく分かっているので、対策として問題が起きたときにリカバリーする策や、一つの手段だけでなく複数通りの手段を用意することで、安全性を保つように考えているようである。

中島：ISO 26262 では、欠陥があった時のリスクを開発者が分析・推定し、ASIL（Automotive Safety Integrity Level: 安全性要求レベル）として四つに分類している。実際の自動車には、クリティカリーレベルとして ASIL-A～ASIL-D の異なった四つのレベルが混在したシステム（Mixed Criticality System:ミックスド・クリティカリティ・システム）が搭載されており、アーキテクチャ上、クリティカリーレベルを明確に分ける。したがって、機械学習のコンポーネントが入ったような大きなシステムがあったときでも、同じようなことになると思う。クリティカリティという言葉が適切かどうかは分からないので、異なるテスタビリティ（Testability）のサブシステムから形成されたミックスド・テスタビリティ・システム（Mixed Testability System）のような形で分けていかなければいけない。これは、たぶんシステムを作るときの常套の考え方であると思う。

福島：品質に関連した有識者コメントの 11 番は対話システムに関するもの。対話システムも様々なユーザーとの対話から学習して変化していくものだと思う。ただ、対話システムを銀行の窓口やコンタクトセンターに入れる場合、学習で変化してしまうのはとても嫌がられるようである。マイクロソフトのチャットボット Tay が悪意あるユーザーとの対話から偏った学習をし、極めて不適切な応答をしたことが影響している面もある。話す言葉のリストや対話パターンをあらかじめ全部列挙してあって、その範囲でやってくれと言われるそうである。したがって、どんどん学習しながら対話を変えていくようなことは難しいし、技術的にもまだまだ時間がかかるようである。今は、仮名漢字変換での優先度を、あらかじめ用意されているパターンのランキングを用いて変えるぐらいのことを学習している段階のようである。

【機械学習型システム開発まわりの関連研究】

福島：機械学習型システム開発に関して、産業界での問題意識も高まっており、研究テーマとして立ち上がりつつあるが、まだまだ研究者層に厚みが足りない感があるのが現状と思う。ただ、機械学習型システムのソフトウェア開発の方法論や開発技術にフォーカスする研究者は少ないかもしれないが、その周辺に関係する研究は盛り上がりを見せていると思う。この点は、有識者コメントの 6～8 番も関係している。
開発方法論自体の研究ではないが、逆にできたものをテストするような視点からの研究、要するに方法論や品質テスト・限界テストにつながるような研究はある。このような視点からの研究は、ある意味、開発方法論の裏返しであり、限界テスト等により品質を高められることにつながる。このような限界テストのような話になると、最近、盛り上がっているトピックとして入ってくると思っている。
それから、有識者コメントの 7 番では、機械学習型システム開発として、学習工場を提唱している。それは、品質問題に関する研究だけではなく、今日、藤吉先生が非常に苦労していると言っていたとおり、データを集めて効率的にアノテーションするための良い方法論やツールをきちんと作るところも重要な問題である。
また、有識者コメントの 8 番目では、帰納的な深層学習と演繹的な記号推論の統合アーキテクチャは重要な AI 分野の研究テーマであり、切口は違うがソフトウェア開発の帰納と演繹に通じるところがあるので、システム開発方法論に関する非常に基礎的な研究であると指摘されている。このような深層学習と記号推論の統合アーキテクチャを AI 的にどうすべきかという点も、最近、重要性が認識されつつある研究テーマなので、もう少し広げて基礎研究として考えることもできると思っている。

丸山：そのあたりは、すごくおもしろい成果が現れ始めていると思う。それは AlphaGo であり、ディスクリートなモンテカルロ木探索（Monte Carlo tree search: MCTS）と、それをガイドする深層学習との組み合わせの手法を用いている。また、最近、Metamath Proof Explorer Home Page に、定理証明に深層学習を取り入れるというログがあり、定理証明では結局、木探索をやるので、その木探索でどの枝を次に探索するかというガイドをするのに深層学習を使っていた。MCTC のようなことをやって、こういうパターンでは定理証明にたどり着く可能性が高かったということを繰り返し学習した 2 例のみ僕は知っているが、もし難しい問題をある種の多項式に表現できたら、あとは、深層学習をオラクルとして使って問題を解くという一つの問題解決のパターンができるかもしれないと思う。そういうやり方はものすごく強力な考え方であり、まさに有識者コメントの 8 番は良いテーマだと思う。
今回の話で言うと、機械学習と演繹的なものを分けて考えないで、それをうまくミックスするようなデザインパターン、あるいはアーキテクチャパターンと言えるだろう。デザインパターンとしての、問題設定のノウハウがたまってくると、それはすごく強力である。

中島：個人的には有識者コメントの 6 番に期待している。自分が興味を持ってやっているのは、ソフトウェア・テスティングのための変なデータを入れたデータセットを作って、それによってプログラムをテストしようという研究である。未知の欠陥がありそうな経路を実行する入力を得ること（コーナーケース検査）をやっているので、

AI 関係や機械学習関係の論文を見ていたら、データセットそのものに対して何かおもしろいことを考えている人はたくさんいることが分かった。データセットシフト（Dataset Shift）という問題から始める論文や、機械学習ではなくてマシンティーチング（Machine Teaching）という論文もある。これは何だろうかという感じがするが、得たい最適値の学習パラメーターを与えて、それに導き出すようなデータセットを作るという話である。それと同じような手法を実はソフトウェア・テスティングにも使えそうなので、様々なやり方でおもしろくて怪しそうなデータセットを作り出すというのは、おそらく機械学習の基礎的な研究でもあり、システム開発の手法にも使えると思う。

【全般について／結びのコメント】

谷： 前田さんが今回最初に説明した本領域での考え方の重要なポイントをずっと見ていて、感じたことがある。要は、システムオブシステムズ（System of Systems）という割と広大な空間での異種システム連携をやらないといけないという問題設定がある中で、例えば機械学習型システム開発では、特定の領域に絞れば様々な課題が出てくる。そのような課題を解決するためには、限界テストや決定論的なものと組み合わせることで対応可能だが、実際にオープンシステムである社会全体に社会実装して実証してみようと言われてしまうと、急にデータセットとして何を用意すればよいのだろうか、あるいは全体として、どのように丁度いいところを見つけることができるのか、とすごく不安になってしまう。もちろん、それなりに科学・技術としてアプローチできると思っているけれども、もともとの領域とのずれが少し心配になってくる。

前田： 我々は、プラットフォーム的な見方をしている。したがって、縦軸として、ただアプリケーションだけをやるというのではなくて、アプリケーションの技術の中で、どれだけ異種システム連携等のイメージに合うようなものを横軸展開できるかという可能性は少なくとも示しつつ、アプリケーションを提案していただくことも可能だし、最初からアーキテクチャ的なご提案も可能であると考えている。2017 年度の公募時にはきちんと言えていなかったので、提案が分かれてしまった原因だったかもしれない。

谷： そうだとすると、今回の公募では、割と縦軸で提案してもよいと思ってよいか？

前田： まだ技術的にもやわらかいので、具体的なアプリケーションを想定した技術開発を提案していただきたいと思う。ただ、その場合でもアプリケーション独自の話だけで固めるのではなく、横軸展開も意識してほしい。

谷： 了解。最終的には、科学・技術として横軸展開できるものを作りたいと思う。

藤吉： 一つだけ、深層学習に特化して言うが、どのようなデータでどのように学習してきたかを再現できるかは、結構、品質にも関連する。そういった仕組みというのはまだないし、丸山さんのところで紹介していただいたモデルのフォーマット標準化は、たぶん、できたものに対する標準化である。したがって、どのようなデータでどのように学習して出てきたものであるかが分かるトレーサビリティ的なもの、それがデータまで含めて（例えば、どこの畑でどうやって出てきてできた野菜かが分かる

ような形で）トレース可能であることが重要だと考える。そういうのは、もしかしたら、この中の技術課題としても入っているとすごくおもしろいし、それも標準化の基準の一つの仕組みになるのではないか。

福島：それをベースに再利用性の評価もできる。

藤吉：そうすると、ウォーターマーキング的なものをネットワークモデルの中に埋め込めば、どこかで勝手にファインチューニングによって改ざんされたとしても、後で知ることができる。もしくは、もともと学習に使用したデータ自体にシステムをだますようなものが含まれていてファインチューニングにより消失しても、改ざんされたことが後で分かる。このようなことは、他の分野でやられているかもしれないが、深層学習にもおそらく活用できるのではないか。これは、ぜひ研究すべきテーマだと思う。

中島：今の話は、その学習システムがどのような履歴でどのようなことを学習してきたかという「学歴」（これは産総研の関口さんから聞いた言い方）を改ざんできないようにする技術で、非常におもしろい。

佐藤：深層学習をするのに、インフラも電気代も結構値段が高いので、ビジネスになかなかなりづらい。電気代とコンピュータを安くしてほしいというのがビジネス側の願いだろう。そのことを置いておくと、期待としては丸山先生のアセスメント、中島先生のスカルセッション、谷さんの構想の部分、そこをうまく進める技術を何か作れないのかというのが、ビジネス側としてはすごく切実な問題である。そこを上手くやれるようなフレームワークや、そこの技術が何かあると本当にありがたい。

谷：どちらかというと、システムエンジニアリング（SE）における要求工学に近い話。方法論があったとしても、この 10 を覚えておけば良い SE のコツみたいなものを超えられるかどうか。学問としてはありうるし、我々も欲しい。それに関しては、全て同意するが、なかなか、その研究が評価されにくいという部分もある。

丸山：でも、それが日本の弱さを象徴しているような気がする。システムエンジニアリングの弱さというのが結局のところ、今の産業の衰退を招いているのではないか。そういうものに対して文部科学省がお金をつけないというのは大きな問題だと思う。本当にそういうことに対する価値が理解されないのは大きな問題ではないか。

前田：私自身はシステム屋なので、本当にそういうことをやりたいと考えているが、なかなかピタッとする提案が少ないというのが現状であり、どうしたものか。

丸山：そこはよく考えていただくとして、もし今の文部科学省的なプロジェクトの立て方でいくのであれば、僕は個人的には二つある。一つは、先ほどの有識者コメントの 8 番、帰納的・演繹的システムの混ぜ方としてのアーキテクチャパターンに関する研究であり、もっと整理されてもいいような気がする。例えば、機械学習が出した結果をルールベースで整理するとか、逆に訓練データセットの健全性をルールベースでチェックするとか、いろんな組み合わせのパターンがあると思う。それぞれ、一長一短があるので、そういうところが研究対象になれば非常に有益である。でも、研究プロジェクトとして本当に最右翼なのは、中島先生がおっしゃったような、品質評価基準の世界標準を日本が作るための研究であり、国家的プロジェクトとして取り組むというのもアリと思う。

前田：それは経産省的な研究テーマかもしれない。JST 的なテーマという制約もあって、2017 年度の応募は、ほとんどがアカデミアからのものだった。でも、実装を考えたところに企業の人を早目に巻き込まなければいけないという面もあり、未来社会創造事業の 2017 年度の問題設定の仕方が難しかった。今日のワークショップでの議論は逆にアカデミア的ではないニュアンスが強いと私は感じた。標準化とか、システムエンジニアリングのご指摘があったが、なかなか大学の先生方でそこに正面切って取り組もうと手を挙げる方はあまり多くないと想像している。2018 年度の公募テーマとしてこの分野を取り上げるには、そのあたりをどういう仕立てにして、どのような公募テーマにするか、今日の議論内容をよく振り返って考えていきたい。個別にご意見を伺う機会があると思うので、その際はご教示よろしくお願いしたい。

木村：私はこの分野は特に専門ではないので、今日は私自身も勉強になることが非常に多かった。最後のほうの議論では、文科省的あるいは経産省的という話があったが、このような分野はもはや単にアカデミア的な技術だけというものではない。社会実装と言った瞬間に、技術と社会の影響を与える側と受ける側の二つを考える必要がある。それを考慮し分かった上で、このようなプロジェクトを進めていかなければならないと私は思う。それが今、日本ではできていないので、停滞しているような気がしている。社会実装に関する人材交流も含めて改善の余地が多大にあり、ファンディングの制度的なところを大胆に一歩踏み出す時期にきていると考えている。我々が、単にシステム・情報という枠の中から、もう一歩踏み出してみるという気概を持って、今回このようなテーマでワークショップを開催した。プロジェクトの性格付けや起こし方に関しても、もう少し緩くフレキシブルに考えてもいいのではないか。それが社会の今の要請であり、期待であると私は思っている。ぜひ今後とも皆様にご協力をお願いしたい。

4. まとめ

前述したように、本ワークショップでは、次の 3 点を目的とした。

① 機械学習型システム開発へのパラダイム転換における課題の全体像をつかむ
② それらの課題に対する取り組みがどこまで進んでいるか、状況を把握する
③ 取り組みを加速･強化するための方針･施策についてアイデアを集める

これら 3 点を中心に、ワークショップでの議論の結果を簡単に振り返る。

①にあげた課題の全体像については、システム開発のフローの視点で整理したものを図 1.4 に、システムの構成要素の視点で整理したものを図 1.5 に示した。今回の発表者のヒアリングを含む事前調査に基づいて作成したものであり、妥当な整理の仕方になっていると受け入れられたものと考えている。ここには、多数のチャレンジングな技術課題がリストアップされている。これらの課題は、アドホックな対処でしのげるものではなく、基本的な考え方のレベルから基礎研究に取り組んでいくことが必要である。

機械学習型システム開発に従来のソフトウェア工学的なアプローチが使えない理由として大きいのは、機械学習を用いて作られたシステムの動作は、原理的に「100%の保証」はできないからである。これは機械学習が本質的に確率的であり、解消できない不確かさを持つということが根本にある。

したがって、機械学習型システムの品質管理には、これを前提とした新しい考え方が必要になる。そして、このような性質を持つシステムの社会実装やシステム開発の受託契約等においても、従来とは異なるタイプのものであることが受け入れられるようになるには、何が必要かを考えていく必要がある。

②にあげた取り組み状況については、システム開発契約や製造物責任法（PL 法）に関わる懸念も含めた品質管理の問題を中心に、産業界での問題意識は非常に高まっている。しかし、システム開発の現場においてはまだ解決策は見えておらず、これを解決する技術開発に対する強いニーズがある。

このような市場や産業界の問題意識の高まりに呼応するように、2017 年はソフトウェア工学系の研究コミュニティにおける取り組みが進み始めた。例えば、2 月 3 日に開催された情報処理学会ソフトウェアジャパンでの招待講演、12 月 14 日に開催された JEITA ソフトウェアエンジニアリング技術ワークショップでの基調講演をはじめ、問題の指摘や解決に向けたアプローチの提唱等が盛んになされるようになってきた。

一方、AI 研究コミュニティの側では、機械学習型システム開発の方法論そのものへの取り組みがまだ手薄のように思える。しかし、機械学習型システム開発に関わる課題を別な側面からとらえたような重要な研究課題への取り組みが進んでいる点には注目したい。例えば、深層学習ベースの画像認識を騙す Adversarial Examples と呼ばれる攻撃のような深層学習の脆弱性に関する研究や、深層学習の解釈性（ブラックボックス問題）に関する

研究は、機械学習型システム開発の品質管理のための基礎研究になる。また、深層学習のような帰納型システムと記号推論のような演繹型システムを統合したアーキテクチャも、AI 研究において重要性が増しているが、機械学習型システム開発という面から見れば、Bigdata-defined と Software-defined の統合のための基礎研究になる。

また、ビジネスの現場では、大量のアノテーション付き学習データを用意しなくてはならないことがボトルネックになっており、データの収集・アノテーション付けの効率化ツールや、少量のアノテーション付きデータから効率よく学習する方法・技術等も、機械学習型システム開発の効率化のため有用である。

③にあげた取り組みの進め方や方針・施策については、まず、この課題への取り組みは、ソフトウェア工学分野と AI 分野のクロスした新しい研究分野を作るものになるが、クロスした分野の研究者群はまだ少ない。両分野の交流とクロス領域の人材育成が重要である。

また、技術課題以外に、制度・ガイドライン面の整備も重要である。産業技術総合研究所では、機械学習型システムの品質基準の検討を開始したと聞いている。PL 法、リコール制度、IEC や ISO 等の安全性評価基準等を考慮し、新しい枠組み作りで世界に先行できれば、産業界の活性化・競争力につながる。

引き続き検討を深め、戦略プロポーザルをまとめていきたいと考えている。

付録

付録1 ワークショップ開催概要・プログラム

【開催日時】2017 年 12 月 8 日（金）13:30～18:00

【開催場所】TKP 市ヶ谷カンファレンスセンター　カンファレンスルーム 7C

【プログラム】

総合司会：茂木 強（JST 研究開発戦略センター フェロー）

13:30～13:40　開催挨拶
木村 康則（JST 研究開発戦略センター 上席フェロー）

13:40～14:00　開催趣旨説明
福島 俊一（JST 研究開発戦略センター フェロー）

発表（発表 20 分＋質疑 10 分）
機械学習型システム開発における課題と課題に対するアプローチや取り組み事例等

14:00～14:30　機械学習工学に向けて
丸山 宏（株式会社 Preferred Networks 最高戦略責任者）

14:30～15:00　ソフトウェア工学×機械学習
～ディペンダブルな機械学習ソフトウェア・システム開発に向けて～
中島 震（国立情報学研究所 情報社会相関研究系 教授）

（10 分休憩）

15:10～15:40　AI を活用した社会価値創造の取り組みと今後の課題
谷 幹也（日本電気株式会社 セキュリティ研究所 所長）

15:40～16:10　機械学習型システム開発へのパラダイム転換 －MPRG の研究紹介－
藤吉 弘亘（中部大学 工学部 ロボット理工学科 教授）

16:10～16:40　機械学習型システム開発のビジネス的課題
佐藤 洋行
（株式会社ブレインパッド
マーケティングプラットフォーム本部 副本部長
チーフデータサイエンティスト）

（10 分休憩）

16:50～17:05　未来社会創造事業の次年度公募テーマに向けた問題意識
前田 章（JST 未来社会創造事業「超スマート社会」領域総括）

17:05～17:50　総合討論
ファシリテーター：福島 俊一
（JST 研究開発戦略センター フェロー）

17:50～18:00　まとめ・閉会挨拶

【参考資料】

- 丸山 宏「機械学習工学に向けて」
  日本ソフトウェア科学会第 34 回大会講演論文集、2017 年 9 月
  http://jssst.or.jp/files/user/taikai/2017/GENERAL/general6-1.pdf

- 中島 震「データセット多様性のソフトウェア・テスティング」
  日本ソフトウェア科学会第 34 回大会講演論文集、2017 年 9 月
  http://jssst.or.jp/files/user/taikai/2017/FOSE/fose3-2.pdf

- 福島 俊一、藤巻 遼平、岡野原 大輔、杉山 将
  「ビッグデータ×機械学習の展望：最先端の技術的チャレンジと広がる応用」
  情報管理 Vol. 60, No.8, 2017 年 11 月
  https://www.jstage.jst.go.jp/article/johokanri/60/8/60_543/_pdf/-char/ja

付録2 参加者一覧

【発表者】

丸山 宏	株式会社 Preferred Networks 最高戦略責任者
中島 震	国立情報学研究所 情報社会相関研究系 教授
谷 幹也	日本電気株式会社 セキュリティ研究所 所長
藤吉 弘亘	中部大学 工学部ロボット理工学科 教授
佐藤 洋行	株式会社ブレインパッド マーケティングプラットフォーム本部 副本部長 チーフデータサイエンティスト

【府省関係者】

蓬莱 史昭	文部科学省 研究振興局 参事官（情報担当）付 研修生

【科学技術振興機構（JST）】

木村 康則	JST 研究開発戦略センター 上席フェロー
福島 俊一	JST 研究開発戦略センター フェロー
茂木 強	JST 研究開発戦略センター フェロー
藤井 新一郎	JST 研究開発戦略センター フェロー
的場 正憲	JST 研究開発戦略センター フェロー
山田 直史	JST 研究開発戦略センター フェロー
前田 知子	JST 研究開発戦略センター フェロー
前田 章	JST 研究開発改革推進部 運営統括
林部 尚	JST 研究開発改革推進部 調査役
奥山 隼人	JST 研究開発改革推進部 主査
加藤 愼一	JST 戦略研究推進部 研究評価グループ 主任調査員
宮田 裕行	JST 戦略研究推進部 ICT グループ 主任調査員

■ワークショップ企画・報告書編纂メンバー■

木村　康則	上席フェロー	（研究開発戦略センター　システム・情報科学技術ユニット）
福島　俊一	フェロー	（研究開発戦略センター　システム・情報科学技術ユニット）
嶋田　義皓	フェロー	（研究開発戦略センター　システム・情報科学技術ユニット）
高島　洋典	フェロー	（研究開発戦略センター　システム・情報科学技術ユニット）
坂内　悟	フェロー	（研究開発戦略センター　システム・情報科学技術ユニット）
藤井 新一郎	フェロー	（研究開発戦略センター　システム・情報科学技術ユニット）
的場　正憲	フェロー	（研究開発戦略センター　システム・情報科学技術ユニット）
茂木　強	フェロー	（研究開発戦略センター　システム・情報科学技術ユニット）
山田　直史	フェロー	（研究開発戦略センター　システム・情報科学技術ユニット）
前田　章	運営統括	（研究開発改革推進部）
林部　尚	調査役	（研究開発改革推進部）
奥山　隼人	主査	（研究開発改革推進部）

※お問い合せ等は下記ユニットまでお願いします。

CRDS-FY2017-WR-11

俯瞰ワークショップ報告書

機械学習型システム開発へのパラダイム転換

平成 30 年 3 月　March 2018
ISBN 978-4-88890-584-8

国立研究開発法人　科学技術振興機構　研究開発戦略センター
システム・情報科学技術ユニット
Systems / Information Science and Technology Unit
Center for Research and Development Strategy
Japan Science and Technology Agency

〒102-0076　東京都千代田区五番町 7　K’s 五番町
電　　話　03-5214-7481（代表）
ファックス　03-5214-7385
http://www.jst.go.jp/crds/

ISBN 978-4-88890-584-8