

3.10 節 「AI の構築・運用」 詳細資料

(2024 年 8 月 30 日版)

映画に出てくるような自律した意識をもつ AI はまだ実現されていない。しかし近年、深層学習などの機械学習で高度な判断や予測ができるようになり、これを AI とよぶようになってきた。この意味で、現時点では AI の開発とは機械学習の開発とほぼ同義である。本節はこの開発プロセスを概観する。

A.3.10.1 AI の開発

AI, すなわち機械学習のモデルを作る作業の流れを図 A.3.30 に示す。AI の開発では、この 1 から 7 のステップを順番に実行していくことになる。モデル化の対象を決め、データを集め、モデルを学習し、その性能を評価し、運用していく、という流れである。以下の項では、これらの項目について説明する。

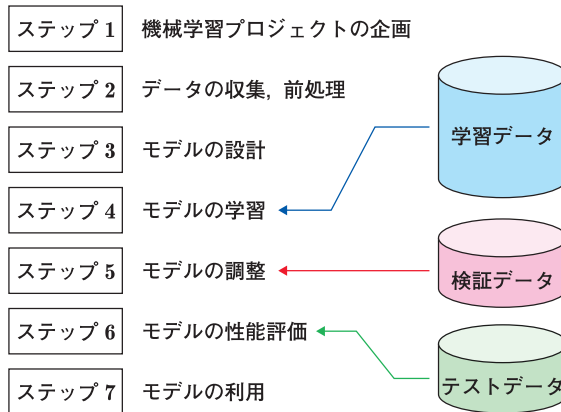


図 A.3.30 AI・機械学習モデルを作るステップ

ステップ1 機械学習プロジェクトの企画

まずは、解決したい課題は何かを決めよう。近年では、企業で使われるデータはほとんどデジタル化されてコンピュータの中に蓄積され、ネットワークでアクセスできるようになっている。営業や財務や流通や人事や製造、すべての部署のデータがそうになっている。その中には、テキスト、画像、音声といったマルチメディアデータも含まれている。こういったデータを使って、もっと業務を効率化したり、新しいサービスを提供したり、顧客の満足度を上げたりしたいと思うのは自然なことである。

機械学習を使って何を実現したいかを決めたら、本当にそれが可能か、データの面から検討してみよう。たとえば、次のような検討事項があるだろう。

- 手持ちのデータに説明能力がありそうか？
 - ◇ 製造装置の異常予測： 利用可能なセンサー情報が異常と関係ない
 - ◇ 効率的な配車： データに不定期イベントの情報が入っていない
 - ◇ 顧客の感情識別： 表情や声ではわからない静かに怒るお客
 - ◇ 需要予測： 一時的な流行や個別の事情などをモデル化するにはデータ不足
- 現場につかってもらえるか？
 - ◇ 前日に需要が予測できても、仕入れは数日前に終わっている
 - ◇ 自動認識の精度が完全ではなく、手作業による修正が大変めんどろ
 - ◇ ウェアラブルセンサーの充電など、現場の仕事が増える
 - ◇ 「面白いね」以上の価値がない

機械の異常検知をおこなうと決めたとしても、機械のセンサーが異常に関係ないデータしか出力しないことがある。言い換えれば、データに説明能力がない、という状況である。その場合には、その企画をあきらめるか、あるいはセンサーの追加を検討しなければならないだろう。データに説明能力がありそうかどうかは、データの可視化によってある程度は推測することができる。

実現したいことと使用するデータに目途がついたら、その企画を実装する価値があるかどうかをチェックしよう。良い機械学習モデルが作れそうなら何で

もやるべきかという点、そうではなく、仕事の現場に使ってもらえなければ意味がないのである。現場に導入されたときに、かえって現場の手間が増えたり、具体的なメリットが人間の勘を超えていなかったりということがありうる。もし、そのような状況が予想されれば、この段階であきらめておくことが賢明である。

ステップ 2 データの収集、前処理

次のステップは、データを集めること、あるいは既に蓄積されているデータを整理・整形して機械学習用に用意することである。

図 A.3.31 の表は、ある工場の機械に関するデータのサンプルであり、1 行が 1 件のデータである。このサンプルには、複数の項目、すなわち多次元の項目データが含まれている。これらは、センサーや機械などによって自動的に取得された値や区分であることが多いだろう。さらに、人間が判断した情報や外部から集めた情報を追加しておく。これをアノテーションとよぶ。たとえば、こちらの例では、取得されたセンサー情報に、稼働状況のアノテーションを追加している。そして、それらを用いてモデルを学習すれば、その稼働状況を推定することができるようになるだろう。所見など自由なコメントも、分析を進めていくうちに、必要性が後からわかることもあるので、この段階では情報を潤沢に入れていくことが重要である。

◆ 集めるのは「**多次元データ**」 + 「**アノテーション**」

センサー1	センサー2	センサー3	センサー4	稼働状況	装置型番	所見
12.0	50.0	0.250	4.90	正常	A3850	バリ多い
10.8	60.6	0.309	5.75	エラーコード A02	A5820	目視確認
10.8	65.1	0.312	5.21	正常	A3850	
49.6	35.0	0.414	7.34	エラーコード B03	A5830	アクチュエータ交換直後

◆ 機械学習では、一般に、データの次元（項目）数が大きいほど、大量のデータが必要

「このような状態のときに、こんな思いがけない結果となっている」ということをモデル化したい

- 稀な状態は多数の項目の組み合わせで表現されることに留意。
- 大量のデータがあれば、その状態を被覆できる。

図 A.3.31 データの収集

一般に機械学習に使用するデータは多ければ多いほど良いといわれている。特に、学習に使用する項目の数が大きいほど、大量のデータが必要である。

たいていの場合、収集した生のデータには、さまざまなエラーが含まれている。そのため、機械学習を問題なく実行できるようにデータを整形しておくことが必要である。この作業を**クレンジング**または**クリーニング**や**データ研磨**とよぶ。

たとえば、図 A.3.32 の入場者数のデータを見てみよう。火曜日のところには数値が入っていない。木曜日のところには、ありえないくらい大きな値が入っている。このような**欠損値** (missing value) や**異常値** (abnormal value) は、同じような週の同じ曜日のデータから類推した値で置き換える、あるいは、利用するデータから除外する、といった作業をおこなっておくべきである。

次にチェックしなければならないのは、データの区分が途中で変わっていないかということである。たとえば、日本では平成時代に**市町村合併**が大々的におこなわれたので、同じ市町村名であっても、人口などの統計が急増しているように見えることがある。この場合、時系列でデータを観察しても矛盾がないように集計を整理する作業が必要になる。

さらに、テキストデータでは表記の揺れや誤りへの対処が必要になることがある。図 A.3.32 の例のように、「iPhone 10」は「iPhone X」と書くのが正し

◆ データクレンジング

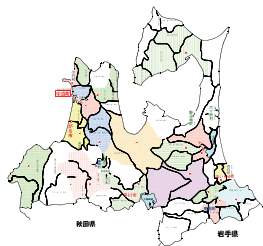
- **欠損値** 値がない
→ 捨てる。あるいは補間値で置き換える。
- **異常値** ありえない値
→ 捨てる。あるいは補間値で置き換える。
- データ区分の途中変更
例) 市区町村の合併
- 誤記、表記の揺れ
例) iPhone 10 と iPhone X

◆ 匿名化

- 個人情報などへの配慮
→ 名前、住所、電話番号などを削除または伏字に
例) 市川さん → [姓] さん

ある温泉の入場者数 (曜日別)

月	火	水	木	金	土	日
90		81	10億	100	89	73



青森県の市町村合併

図 A.3.32 データの前処理

地図の出所：国土地理院 全国都道府県別・市町村合併新旧一覧図 (平成 15 年以降)

<https://www.gsi.go.jp/common/000049738.pdf>

い表記である。

最後に、**匿名化**の作業をおこなう。名前、住所、電話番号などの個人情報は、削除したり、伏せ字に置き換えておく。主な目的は、個人情報を分析者の目に触れないようにすることであるが、誤って個人情報を機械学習のモデルに覚えさせないようにすることも重要である。匿名化はテキストデータが対象となることが多いが、音声データや画像データであっても配慮する必要がある。

クレンジング済みのデータは、一部をモデルのテストデータとして取り分けておこう (3.3 節参照)。通常は、取り分けたデータに偏りがないようにデータ順をシャッフルしてから取り分けるようにする。あるいは単純にランダムイズするだけでなく、性別、年代、機種といったデータ区分について、割合のバランスをとるように配慮することもある。残りはモデルの学習に用いるデータとなるが、それをさらに少量だけ取り分けて、検証データとする。その残りが学習データとなる。取り分けたテストデータや検証データのサイズがあまり大きいと、学習データが少なくなってしまうので、注意しよう。

次におこなうのは、データをモデルへ入力するのに適した形に変換することである。3.6 節で述べたように、モデルへの入力の特徴量とよぶ、データを良い特徴量へ変換するために、以下の前処理をおこなうことが多い。

- 標準化
- 変数選択
- 次元削減

標準化については、1.4 節で学習した、多次元データのそれぞれの項目について、項目ごとの平均を引き算し、項目ごとの標準偏差で割り算しておくことである¹⁰⁾。

変数選択は、良い特徴量とするために必要となることがある。特徴量ベクトルの次元は無駄に多すぎないほうが良いのである。たとえば、機械の似たような場所にセンサーを 2 つ設置して、同じ事象を観察したとしても、情報は 2 倍にはならないのは自明である。むしろ、多重共線性といって、モデルの性能が下がる現象がみられるのである。特に学習データの量が少ないケースでは、必

¹⁰⁾ 標準化は機械学習の方式によっては必要ない。

要十分な変数のみを残した次元数の少ない特徴量とすることが望ましい。

次元削減は、1.6 節の web 資料にある主成分分析が代表例である。冗長な変数を削除する代わりに、複数の変数を合成して、あらたな変数を少量だけ作り、結果的に次元が削減された特徴量を生成するものである。線形判別分析 (LDA) も次元削減に使われる。

ステップ 3 モデルの設計

特徴量データの準備ができれば、モデルの設計をおこなう。使用するデータには、判断の材料となる観測データが含まれているだろう。それらが、何か予測したい数値や分類とセットになっているだろうか？

以下のリストに従って、作成したい機械学習のモデルがどのタイプに合致するかを考えよう。

- データが、それを使用して予測したい数値や分類とセットになっている
 - ◇ たとえば、異常検知問題で、正常状態と異常状態の 2 分類のデータを用意できる
 - 教師あり学習 (分類問題)
 - ◇ たとえば、混雑度 (%) を数値で予測したい
 - 教師あり学習 (回帰問題)
- 予測したい数値や分類とセットになっていない
 - ◇ たとえば、異常検知問題で、正常状態のデータしか用意できない
 - 教師なし学習
 - ◇ 異常検知問題では、正常状態からの逸脱の度合いをチェックする

解決したい課題によっては、数値や分類を予測するのではなく、データを生成する機械学習モデルが必要になることがある。たとえば、日本語の単語列を入力して、英語の単語列の翻訳を得たい場合には、系列変換モデルというタイプが必要になるだろう。たとえば、写真画像を入力して、油絵タッチの画像を出力したいという場合には、敵対的生成ネットワークなどのモデルが必要にな

るだろう。

教師あり学習の分類問題では、以下のリストに例示した機械学習の手法を適用することができる。これらの手法には一長一短があり、どれを採用すべきかは一概にはいえない。収集したデータの量や特性、解決したい課題の内容に応じて選択して欲しい。もし、データが大量にあるのであれば、迷わずにニューラルネットワークモデルを適用するのも 1 つの方策である。

- ロジスティック回帰モデル
- 単純ベイズモデル
- ニューラルネットワーク (深層学習) モデル
- 決定木 (分類木) モデル
- サポートベクターマシン (SVM) モデル
- 最近傍モデル

教師あり学習の回帰問題では、以下のリストに例示した機械学習の手法を適用することができる。どの手法を採用すべきかは一概にはいえないが、学習データの量が少ない場合には、線形回帰モデルを使用するのが無難であろう。また、Python や R を用いて自分でプログラミングをおこなう場合には、採用する手法に対応したライブラリが入手可能かのチェックは欠かせない。

- 線形回帰モデル
- ニューラルネットワーク (深層学習) モデル
- 決定木 (回帰木) モデル
- サポートベクター回帰 (SVR) モデル

教師なし学習では、異常検知など分類問題を構成することができる。以下のリストに例示した機械学習の手法を適用することができるが、どの手法を使用したとしても、正常時の状態をモデルに教えることが、学習時の処理となる。推論時は、テストデータを入力し、正常状態からの逸脱をチェックし、大きく逸脱していれば異常と判断する。

- MT 法
- オートエンコーダ

- 1 クラス SVM
- カーネル密度関数法

ステップ4 モデルの学習

特徴量の準備ができ、どのようなタイプのモデルとするかの設計が済んだら、次は学習の実行である。

学習を実施するにあたっては、どのようなツールを使うかの選択がある。モデルの学習は、Python や R や MATLAB などの言語を用いて、自分でプログラミングをおこなうのが、王道である。機械学習の方式やモデルの構成や学習のパラメータといった設定を細部に至るまで自分で決めることができるからである。これらの言語では、ライブラリが豊富にあり機械学習のほとんどの手法を利用することができる。また、ライブラリの中には、計算用の **GPU** (GPGPU) に対応したものもあり、特にニューラルネットワークの学習処理を高速化することができる。プログラミングには、Windows や Linux といった手持ちのパソコンを使用してもよいし、Google Colaboratory などのクラウドの開発環境を使用してもよい。

他の方法として、JMP, SAS, Stata, SPSS などの商用のアプリケーションを使用するという選択肢もある。基本的には、機械学習の方式を選択して、データを与えれば、自分でプログラミングをおこなうことなしに、モデルの学習やデータの分析をおこなうことができる。

また、商用の**クラウド AI サービス**を利用するという選択肢もある。こちらは、学習データをクラウド AI へアップロードすることによって学習を実行したり、音声認識や画像認識などの学習済みのモデルをそのまま利用したり、モデルを手持ちデータで再学習して利用したりするものである。機械学習の手法については既に決められていて、自分でプログラミングする自由度はないが、クラウド上に機械学習のモデルが作成されるので、それを利用するサービスを世の中に公開しやすいというメリットがある。多くのクラウド AI サービスは、webAPI という形式で機能を提供している (2.3 節参照)。これは、ざっくり言うとインターネットの web ページを閲覧したり、ファイルをアップロードしたりする手順と同じような仕組みで機能呼び出しができるようにしたものであり、Python

や Java で書かれたプログラムから容易に機能呼び出すことができる。商用のクラウド AI サービスとしては、Google Cloud Platform, IBM Watson, AWS, Microsoft Azure などがあげられる。

モデルを学習する際に、特にニューラルネットワークモデルで問題になるのが、**不均衡データ**の扱いである¹¹⁾。

たとえば、異常検知のモデルで、「正常」と「異常」の2つの分類クラスがあったときに、「異常」クラスの学習データが、「正常」クラスのそれよりも大変に少ないということはよくあることである。その場合、推論時に、なかなか「異常」クラスが出力されないという問題が起きることがある。

この問題に対処するには、クラスごとの学習データの数を実質的に揃うようにするとよい。たとえば、アンダーサンプリングといって、学習データの多いクラスについて、学習時にデータをランダムに間引いて少なくすることがおこなわれる。あるいは、オーバーサンプリングといって、学習データの少ないクラスについて、学習時にデータを繰り返し用いて増量することがおこなわれる。データを繰り返し用いるのではなく、学習時の重み係数をその分だけ大きくしても同じ効果を期待できる。あるいはデータとデータの間を補間する形で人工的にデータを生成して増量するという方法を適用することもある。

ニューラルネットワークモデル、特に層を重ねた深層学習モデルで重要になってきている手法が、**ファインチューニング** (fine-tuning) と**転移学習**である。

一般に深層学習モデルは、層が多く積まれていて中のパラメータの数が多い複雑なモデルほど、性能が高いといわれている。しかし、そのような複雑なモデルを学習するには大量のデータが必要となる。一方で、自分の問題を解きたいときには、少量のデータしか用意できないことが多い。そこで、非常に性能の高い事前学習済モデル (pre-trained model) を入手して、それを、自分の少量の手持ちデータで、追加的に学習操作をおこない、自分用のモデルを作成することがおこなわれる。これをファインチューニングとよぶ。

転移学習は、前述のファインチューニングと似ているが、下層すなわち入力層に近い数層のみを、学習済みモデルから拝借するやり方である。一般に、深

¹¹⁾ 不均衡データが問題にならない機械学習の方式もある。

層学習の下層は、入力データから判断のための特徴を取り出す働きがあるといわれている。たとえば、画像データの場合、下層では物体の輪郭を取り出すような、基本的な操作となる。一方で、上層になるにしたがって、推論の根拠となるような具体的な情報の抽出がおこなわれる。したがって、たとえば、利用可能なデータが少ない少数言語の音声認識を作る際に、非常によく学習された英語の音声認識の音響モデルを入手してその下層をコピーし、少数言語の音響モデルを学習させるときの初期値として利用することができる。

ステップ5 モデルの調整

モデルの調整には2つの側面がある。1つは、モデルの選択である。背景として、特にニューラルネットワークモデルでは、層やユニットの少ないシンプルな構成から、それらが多い複雑な構成まで、いろいろな構成が考えられるが、どれが一番性能の良いモデルになるのか、事前にはなかなか予想できず、結局、多くの種類のモデルを学習してしまうことが多いのである。また、1つの構成のモデルであっても、学習は反復的におこなわれるので、反復回数分のモデルが存在するのである。そこで、検証データを用いてテストをおこない、性能が一番良いものを採用する、ということをおこなう。

モデルの調整のもう1つの側面は、モデルの外から調整できる閾値や重みなどの調整である。たとえば、画像データを入力して、それが「犬である」か「犬でない」かを分類するモデルの場合、「犬」判定の閾値を小さく調整すれば、「犬」と判定されやすいモデルにチューニングすることができる。どのような閾値とすべきかは、やはり、検証データを用いてテストをおこない、性能が一番良くなる値を採用すればよいわけである。

ステップ6 モデルの性能評価

モデルの学習とその調整が済んだら、テストデータを用いて性能を計測する。以下の内容は、3.3節で学んだ通りである。

分類問題では、テスト結果の混同行列を作り、適合率と再現率を算出して評価しよう。

適合率と再現率のどちらを重視するかは、対象とする課題による。たとえば

ガンの診断であれば、見落としは困るので適合率よりも再現率を重視するだろう。そうではない一般的な問題であれば、 F 値としてバランスをとった値で評価するのがよいであろう。また、正常・異常を判定する問題の場合、設定する閾値によって、適合率と再現率のバランスは変わるので、純粹にモデルの良し悪しを判定するためには、ROC 曲線を描いて AUC で評価するとよいだろう。

回帰問題では、テスト結果の予測数値が得られるので、正解の数値と比べればよい。すなわち、テストデータを 1 件入力するごとに、両者の差を計測し、絶対値をとる、あるいは 2 乗値を計算する。全データについて平均をとれば、平均絶対誤差、あるいは平均 2 乗誤差という指標を計算できる。これらの値が小さいほうが、モデルの性能が高い、ということになる。

ステップ 7 モデルの利用

ここまでで、モデルの学習が完了し、性能も計測された。性能が満足できるものであれば、実際にそのモデルをシステム上に実装し、企業や社会の中で使ってもらおう。ただし、機械学習のモデル単体では、入力を与えると出力を出すという機能部品にすぎないので、組み込むシステムから呼び出せるように周辺のコードを追加する必要がある。また、システムに応じてプログラム言語やプラットフォームを載せ替えることがある¹²⁾。

モデルを利用するというのは、推論動作であるので、学習動作と比較すれば計算負荷はそれほど高くない。したがって、多くのケースで GPU は必須ではなく、パソコンと Python で実行すれば十分である。ただし、自動運転への応用など高速動作が必要な場合には、それでは不十分となる。Python はインタプリタ言語なので動作は遅い。また、パソコンを自動車に組み込むのも嫌がられる。したがって、Python のプログラムコードと学習されたモデルを、高速動作する形式に変換したり、組み込み型の GPU 機器を利用したりすることがおこなわれる。

¹²⁾ クラウド AI サービスであれば、学習済みモデルは載せ替えの必要はない。システムは web API を通じてモデルの機能呼び出しをおこなうことができる。

新しい流れ：AutoML

ここまで、機械学習のモデルの開発のステップについて見てきた。各ステップでは、いろいろな選択肢があり、作業も複雑であるため、豊富な経験が必要という印象をもったことと思う。実は、最近の新しい流れとして、ステップ3のモデル設計からステップ6のモデルの性能評価までの処理を、全部お任せできるツールが普及し始めている。これは **AutoML** とよばれている。AutoMLを使えば、機械学習の専門家でなくても、機械学習モデルを用意に開発することができそうである。

A.3.10.2 AI の社会実装

日常生活の中で AI は普及しつつある。たとえば、スマホのカメラで身の回りのものを撮影すると、その物の名前が出てくるとか、日本語をいろいろな国の言葉に自動で翻訳してくれるアプリとか、あるいは自動で作曲や作画をしてくれる AI などを思いつくだろう。これらは個人から見える AI である。また、企業や社会の現場でも AI の実装が進んでいる。これらは個人からは見えにくいところにある AI である。気が付きにくい社会を効率化するための重要な役割を担っている。以下の小節では、それらの例を概観する。読者が今後 AI の社会実装を進める際の参考にして欲しい。

(1) 生産現場で活躍する AI

AI は製造業や農業の生産現場を効率化させるために用いられる。以下に例をあげる。

- 工場出荷検査
 - ◇ 製造した部品に、傷、バリ、汚れはないか？
 - ◇ 視認が難しいヒビや異物の混入はないか？
- 野菜の出荷検査
 - ◇ 曲がったキュウリはないか？
 - ◇ サイズごとに仕分けたい
- 機械の異常検知・予測

- ◇ 温度、圧力、電圧・電流、振動・音などのセンサー情報を用いて、異常の発生を検知
- ◇ 異常が発生する前に予兆を捉えて、部品の交換 → 予防保全

(2) 人や物の流れの効率化のための AI

AI は人や物の流れを効率化させるために用いられる。以下に例をあげる。

- 天気予報を用いて売上を予測し、仕入れ量を調節する
- 顔画像の認証を用いることで、テーマパークの入退場や入出国検査を効率化
- 季節や曜日や天候の情報を用いて、テーマパークの入場者数や高速道路の渋滞を予測
- 呼び出しがありそうな場所・時間に、あらかじめタクシーや救急車を待機させておく
- 宅配便のセンターの業務量を数カ月先まで予測

(3) サービス品質の向上のための AI

AI はサービスの質を向上させるために用いられる。以下に例をあげる。

- レジが無い無人店舗では、顧客が何を買ったかを自動認識
- コールセンターのオペレータの会話を自動的に検査して、正しい応対をしているかを自動でチェック
- 裁判員を補助するために、公判の音声をテキスト化
- 国会議事の音声を速記官の代わりに自動でテキスト化
- 医師を補助するために、医療画像を自動的に診断
- 契約書などの文書が正しく書かれているか自動でチェック
- 新聞記事の要約を、求めるサイズに応じて自動的に作成

(4) 利益向上のための AI

ビジネスの分野では利益を向上させるために AI が用いられる。以下に例をあげる。

- バランスシートや口座の取引情報を用いて、取引先企業の信用を評価

- 衛星で取得した海水温データを用いて、魚群がいそうなところへ漁船を配置
- 水、肥料、気温、日照を制御して農産物の収穫を最大化
- ネット通販の顧客の購買傾向を分析して、購入しそうな物を提案
- 株価や為替のデータを用いて、リアルタイムに自動売買
- 最寄り駅、築年、日当たり、地区、構造、広さなどの情報から、不動産の適正価格を推定

(5) 社会のための AI

社会のための AI は、公共サービス、環境、防犯・安全などの観点から展開される。以下に例をあげる。

- ゴミ収集車に路面撮影のためのカメラを装着して、路面の破損や落下物やペイントのかすれなどを自動検知して道路の保守に活用
- 運転手のいない自動運転バスの運行
- 森林の成長と木材価格を予測し、森林資源の管理を最適化
- 振り込め詐欺の電話を自動検知し警告
- AI を装備したロボットによる自動警備
- 中国では、街のいたるところに防犯カメラが設置され、通行人の顔認識がおこなわれている (天網)

A.3.10.3 AI と DX

ここまで見てきたように AI の活用事例は大変に幅広いものであって、AI は社会を大きく変革する可能性を秘めている。

社会の変革については、デジタルトランスフォーメーション (DX) という言葉がキーワードとなっている。図 A.3.33 に示すように、DX はデータのデジタル化を進め、データサイエンスと IT の力を結合させ、企業のビジネスの仕組みを大幅に効率化したり、新たな価値を作り出したりする取り組みである。AI は、DX のための不可欠な技術要素となっており、これからも社会のインフラに広がっていくだろう。

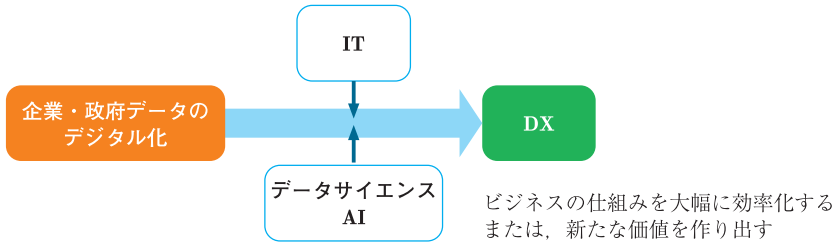


図 A.3.33 AI と DX の関係

課題学習

A.3.10-1 個人宅における時間別の電気の使用量を入力として、その家の住人の数や年齢構成を予測する機械学習モデルを作成することを企画した。どのようなデータをどれくらいの分量を集めるかプランを書け。また、集めたデータをどのように加工し、どのようなタイプのモデルとして学習するのかそのプランを書け。

A.3.10-2 AI を用いた新たなサービスや製品を提案してみよ。これまで誰にも提唱されていなかった案が望ましい。