機械学習とは何か？

背景と概要

◆世界的企業によるAIの取り組み

トップ５社のCEO発言

▶ トップ5社のCEO発言 図表02-5

Google CEO スンダー・ピチャイ （2017 年 5 月 Google I/O '17）	モバイルファーストから AI ファーストにシフトする
Amazon CEO ジェフ・ベゾス （2016 年株主への手紙）	我々は今まさに、明確な力強いトレンドのさなかにある。AI と機械学習だ
Facebook CEO マーク・ザッカーバーグ （2018 年 2 月の Facebook 投稿）	AI を用いた我々のゴールは、よりよいサービスを作るために、Facebook 上のすべてのコンテンツを理解できるようにすることだ（図表02-6）
Apple CEO ティム・クック （2017 年 8 月の決算説明会）	自律的なシステム（autonomous systems）の巨大なプロジェクトを動かしていて、莫大な投資をしている。乗り物もその使い方の 1 つだが、ほかにもさまざまな使い道のある、あらゆる AI プロジェクトの母親のようなものに取り組んでいる
Microsoft CEO サティア・ナデラ （2017 年のアニュアルレポート）	あらゆる開発者が AI 開発者に、あらゆる企業が AI 企業になれるよう、我々の AI に関するケイパビリティを使って特徴あるポジショニングを取り、AI を民主化する

◆トップ企業にみる機械学習の取り組み

・Google
　☞自動運転の紹介動画
　　▶ https://waymo.com/
　☞医療・科学・芸術ほか
　　医療　▶ https://ai.googleblog.com/2017/03/assisting-pathologists-in-detecting.html
　　科学　▶ https://www.blog.google/topics/machine-learning/hunting-planets-machine-learning/
　　芸術　▶ https://experiments.withgoogle.com/ai/ai-duet/view/

・Amazon
　☞無人配達
　　▶ https://www.youtube.com/watch?v=vNySOrI2Ny8
　☞レジ無し店舗
　　▶ https://www.youtube.com/watch?v=NrmMk1Myrxc

・Facebook
　☞SNSコンテンツの理解
　　▶ https://research.fb.com/downloads/detectron/

・Apple
　☞ユーザ体験の向上

・Microsoft
　☞顧客企業の支援

◆機械学習が注目される理由

・注目されるきっかけとなった事例

（１）画像認識

☞画像認識の世界的なコンペティション（ILSVRC：ImageNet Large Scale Visual Recognition Challenge）において、２０１２年にディープラーニング（深層学習）という方法を用いて大幅な画像認識の精度の改善が達成された。
LSVRCでは任意の画像に何が写っているかを判別し、エラー（不正解）率の低さを、 世界中の企業や大学の研究者たちが競い合う。
ディープラーニングの手法を用いたチームは、それまで約50年の人工知能研究の成果で到達した26%のエラー率を、一気に15%にまで改善した。

（２）囲碁

☞2016年にGoogle 傘下のDeepMindの開発した「AlphaGo」が、囲碁の世界チャンピオンであるイ・セドル氏を破った。
https://www.youtube.com/watch?v=WXuK6gekU1Y
AlphaGoには、ディープラーニングだけでなく、強化学習という機械学習の手法が用いられている。
https://ja.wikipedia.org/wiki/AlphaGo

・なぜ機械学習が成果を生むようになったのか？

アルゴリズムの進化	・ディープラーニングや強化学習など、応用可能性と実行力の高い機械学習の手法の進化が続いているため
データ量の増大	・インターネットの発展により、帯域の増強と共に画像、映像、音声、テキストなどさまざまなデータが増大しているため ・企業における業務のシステム化やセンサーの普及により、さまざまな種類のデータ量が増大しているため
計算資源の進化	・コンピューターの処理能力が格段に高まっているため（スパコンの処理能力の向上、GPU、TPU などの開発・普及）
アルゴリズム、データ、計算資源の利用可能性の向上	・オープンソースのライブラリや TensorFlow などのツールにより、機械学習やディープラーニングのアルゴリズムが簡単に使えるようになっているため ・各種のデータソースが整備され、学習のためのデータが用意しやすくなっているため ・クラウドサービスの普及により、高性能の計算資源が安価に使えるようになっているため

◆機械学習ライブラリ・フレームワーク

1．NumPy

☞NumPyの特徴

・数値計算を行うための定番ライブラリの1つで、「ナンパイ」または「ナムパイ」と読む。
・機械学習だけでなく、多言語配列や画像処理・音声処理にも活用できる利用頻度の高いライブラリ。
・数値計算を得意とするライブラリだが、SciPyを追加する事で、さらに高度な科学計算を処理することが可能。

☞NumPyの便利な点

・NumPyは数値計算の中で、特に配列処理能力に優れている。Pythonでも計算することは可能だが、インタプリタ型のプログラミング言語のため、実行速度が遅く、処理に時間がかかる。NumPyはC言語やFortranといったコンパイル型言語で実装されているため、処理速度が高速。
・データ分析において、データを受け渡すためのデータコンテナとしての役割もある。
・NumPyにC言語呼び出しのAPIがある事で、NumPyから外部ライブラリへデータを受け渡すことができたり、逆に外部ライブラリの計算結果をNumPyに戻すことが可能。
・この機能によって、NumPyは既存ライブラリを簡単に呼び出せる、動的インターフェースとしての役割を担ってる。

☞NumPyの利用シーン

・ベクトルや行列といった多次元配列をPythonより早く効率的に処理する必要があるプログラムで利用される。
・大規模のデータ処理に優れているだけではなく、他ライブラリと連結することがある。例えば「Pandas」「SciPy」「Scikit-learn」といったライブラリと連結して使用することが実際の業務で多数発生する。機械学習の世界では、かなり高い確率で使うことになる。

２．Pandas（パンダス）

☞Pandasの特徴

・Pandasは、数表および時系列のデータ操作やデータ構造を変更するなど、テーブルデータを取り扱えるようにするライブラリ。
・機械学習において、大量のデータをAIに学習さるが、最終的なモデル精度をより高めるためには、不要なデータを取り除き、必要なデータを精査する前処理をする必要がある。
・Pandasはその前処理の際に、データセット処理を効率化するために使用する。
・Pandas は重要なコードをC言語で実装しているため、NumPy同様、高速に処理することが可能。

☞Pandasの便利な点

・NumPyで作成されたデータの加工や入出力が可能。
・数値以外のデータ処理を行う場合にPandasを使うと便利。
・SQLと似た操作でデータ加工が行えるためデータベースを触っている人は使いやすい。
・値に対するラベル付けを簡単に行えるため、機械学習の前処理などの工程を効率的に行うことが可能。
・合計や平均だけでなく、分散や標準偏差といった機械学習において必須となる統計量も簡単に処理できる。
・機械学習でよく行われる特定条件のデータを除外する処理も、SQLを書くように記述できる。
・他にもソート、欠損値の補完、グラフの描画等、テーブルデータを取り扱う上で必須のメソッドを多く備えている。

☞Pandasの利用シーン

・一般的な表計算から、統計量の算出、データ整形、csv等のさまざまなフォーマットでの入出力といった、テーブルデータを扱う場合に、その豊富な機能面から利用されるデータ分析に必須のライブラリ。
・金融データを取り扱うために最も適した時系列分析機能をもっているため、金融データ分析アプリケーションにも利用されている。
・機械学習でデータ分析を行う前処理であるデータの読み込みやクリーニングといった作業でよく利用さる。

３．SciPy（サイパイ）

☞SciPyの特徴

・SciPyは、信号処理や統計などの科学計算用のライブラリ。
・語源は Science + Python に由来。
・SciPyではNumpyで行える配列や行列の演算はもちろん、さらに信号処理や統計といった計算ができるライブラリ。
・NumPyよりも高度な数値計算処理を行う場合は、SciPyを利用する。

☞SciPyの便利な点

・SciPyは統計、最適化、補完、積分、線形代数、フーリエ変換、信号処理、画像処理、遺伝的アルゴリズム、ODEソルバ、特殊関数といった、高度な科学技術計算処理をPythonを使って実行可能にする。

☞SciPyの利用シーン

cluster：階層的クラスタリング、ベクトル量子化、K平均 fftpack: 離散フーリエ変換
integrate：数値積分ルーチン、微分方程式ソルバ
linalg：線形代数ルーチン
io:データ入出力
maxentropy：エントロピー分布処理
ndimagemisc：画像処理
sparsemisc：疎行列処理
stats：統計処理、離散分布、連続分布
signal：信号処理ツール
tsci：画像配列操作
weave：PythonにC言語やC++言語を組み込む処理

４．Matplotlib（マットプロットリブ）

☞Matplotlibの特徴

・データをグラフや画像データとして表示することができるブラフ描画のためのライブラリ。
・機械学習では、統計量の可視化や学習経過のグラフ化、画像の出力等の機能が多く利用されている。
・ヒストグラムや散布図を描いたり、JavaScriptを利用してインタラクティブなグラフを生成することも可能。
・Pandasでもデータの可視化は可能だが、Matplotlibを利用する事で更に複雑な表示が可能。

☞Matplotlibの便利な点

・出版用にも使えるほど高品質なグラフを作成することができる。
・画像をPDFやJPEG、GIF等であらゆる形式でエクスポート可能。

☞Matplotlibの利用シーン

・NumPyなど他ライブラリで前処理したデータを可視化する場合によく使われる最も一般的なPythonライブラリ。
・データを可視化する事で、異常値の検出や必要データの変形を発見することができるため、データ分析の現場ではよく使われる。
・機械学習でも、NumPyやPandasと組み合わせ、分析対象のデータにどんな傾向があるのかグラフで表示したり、スコアの変化を描画する等、データを図表で表示させる際に活用されている。

5．scikit-learn（サイキット・ラーン）

☞scikit-learnの特徴

・機械学習全般のアルゴリズムが実装された機械学習の基盤となっている大人気のライブラリ。
・統計学、パターン認識、データ解析の技法が豊富に使うことができるので、特に研究者の間で人気がある。
・NumPyやSciPy、matplotlibと比較してもscikit-learnは、様々な機械学習の実装を簡単に試すことができる。

☞scikit-learnの便利な点

・機械学習全般のアルゴリズムを簡単に実装できる。
・ただしどのアルゴリズムを選ぶべきか検討する際に、チートシートが参考になる。
・データの状況に応じで細かく分岐されたチートシートを活用する事でアルゴリズム選択が容易になる。

☞scikit-learnの利用シーン

・主に分類・回帰・クラスタリング・次元削減という４つの目的に応じて利用される。

分類：
ラベルとデータを学習し、ラベルがわからないデータにラベルを付けること。
例えば、身長体重とシワの数から性別を識別するというようなことに使われるイメージ。

回帰：
数値をデータで学習し、その学習モデルを利用して、数値を予測すること。
例えば、身長体重かとシワの数から年齢を予測するといったことに使われるイメージ。

クラスタリング：
漠然としたデータから近しい特徴を見つけて分類し、その学習モデルを利用して、同じようなデータからグループを見つけ分類すること。
例えば、星を光の強さや大きさ、色、距離、温度といったいくつかの特徴をもとにグループに分類するときに使われるイメージ。

次元削減：
データの次元を削減しデータの可視化、圧縮化を行うこと。
イメージでいうと、３次元である現実世界から写真を撮ることで2次元に落とし込み、状況を判断するようなことをデータでも行うという感じ。

◆ディープラーニングフレームワーク

1．TensorFlow

・TensorFlowはGoogleのGoogle Brainチームによって2015年に開発。
・ディープラーニングフレームワークの中で現在最も人気のあるフレームワーク。
・Googleが開発したフレームワークなので、GmailやGoogle翻訳などに使われている。

主な特徴としては、

・C++とPythonで書かれている
・ドキュメンテーションが充実している
・iOSやAndroidなどのモバイルのプラットフォームでも利用できる
・Tensorboardと呼ばれる、訓練過程を観察できる機能がついている

☞公式サイト： https://www.tensorflow.org/

2． PyTorch

・PyTorchはFacebookの人工知能グループによる開発からスタートし、2016年にリリース。
・上で述べたTensorFlowの人気に最も近い。

主な特徴としては、

・構文がわかりやすい
・研究者の間で人気
論文で発表されたモデルの実際のコードは、PyTorchで書かれたものが比較的見つかりやすい
・pdbやPyCharmなどのデバッガーが使える
動的な計算グラフ（学習速度は落ちるものの、柔軟なモデルの構築が可能）

☞公式サイト： https://pytorch.org/

３．Keras
・KerasはFrancois Chollet氏（現在はGoogleのエンジニア）を中心として2015年に開発。
・現在はTensorFlowに取り込まれ、tf.kerasの形で使われるのが一般的。
・開発者のFrancois Chollet氏自身が執筆したKerasの解説本も出版されている。
主な特徴としては、
・モデルの構築がとても簡単で、初心者にとってもわかりやすい。
・素早く実装できるので、プロトタイプの作成時に便利。
・TensorFlow、CNTK、Theanoという複数のバックエンドをサポート。
👉公式サイト： https://keras.io/

４．Caffe
・Yangqing Jia氏が、カリフォルニア大学バークレー校の博士課程在学中に始めたプロジェクトで、同校のBAIR（Berkeley Artificial Intelligence Research）を中心に開発が行われ、2017年にリリース。
主な特徴としては、
・画像認識などの処理が得意
・開発コミュニティが活発
・動作が速い
👉公式サイト： https://caffe.berkeleyvision.org/

5．Microsoft Cognitive Toolkit

・元々CNTKという名前で2016年にリリースされたが、同年10月に現在の名前に変更された。
・Skypeなどにも使われてる。

主な特徴としては、

・リソース効率が良い
・ONNXフォーマット（ディープラーニングモデルを異なるフレームワーク間で交換するためのフォーマット）を初めてサポート
・巨大データセット処理時のパフォーマンス低下を最小化するためのアルゴリズムが組み込まれているので、複雑マシンで巨大データセットを扱える
・コミュニティはそこまで発展していない

👉公式サイト：https://www.microsoft.com/en-us/research/product/cognitive-toolkit/?lang=fr_ca

6．MxNet

・MxNetは、CMU（カーネギーメロン大学）, NYU（ニューヨーク大学）, NUS（シンガポール国立大学）, MIT（マサチューセッツ工科大学）など、様々な大学からの研究者が協力して開発され、2016年にAWS（Amazon Web Service）によるサポートが発表された。

主な特徴としては、

・高いスケーラビリティ
・モバイルデバイスも対応可
・Python, R, Scala, JavaScript, C++など多くの言語に対応
・研究者の間での人気はあまり高くなく、コミュニティもそこまで発達していない
・命令的プログラムと宣言的プログラムの併用が可能

👉公式サイト：https://mxnet.apache.org/versions/1.7/

7．Chainer

・日本のベンチャー企業である、PFN（Preferred Networks）によって開発。
・2019年12月にその開発を終了し、PyTorchへ移行されることが発表されたが、define-by-runというアプローチを他のフレームワークに先駆けて提唱するなど、後に続くフレームワークにも少なからず影響を与えた。

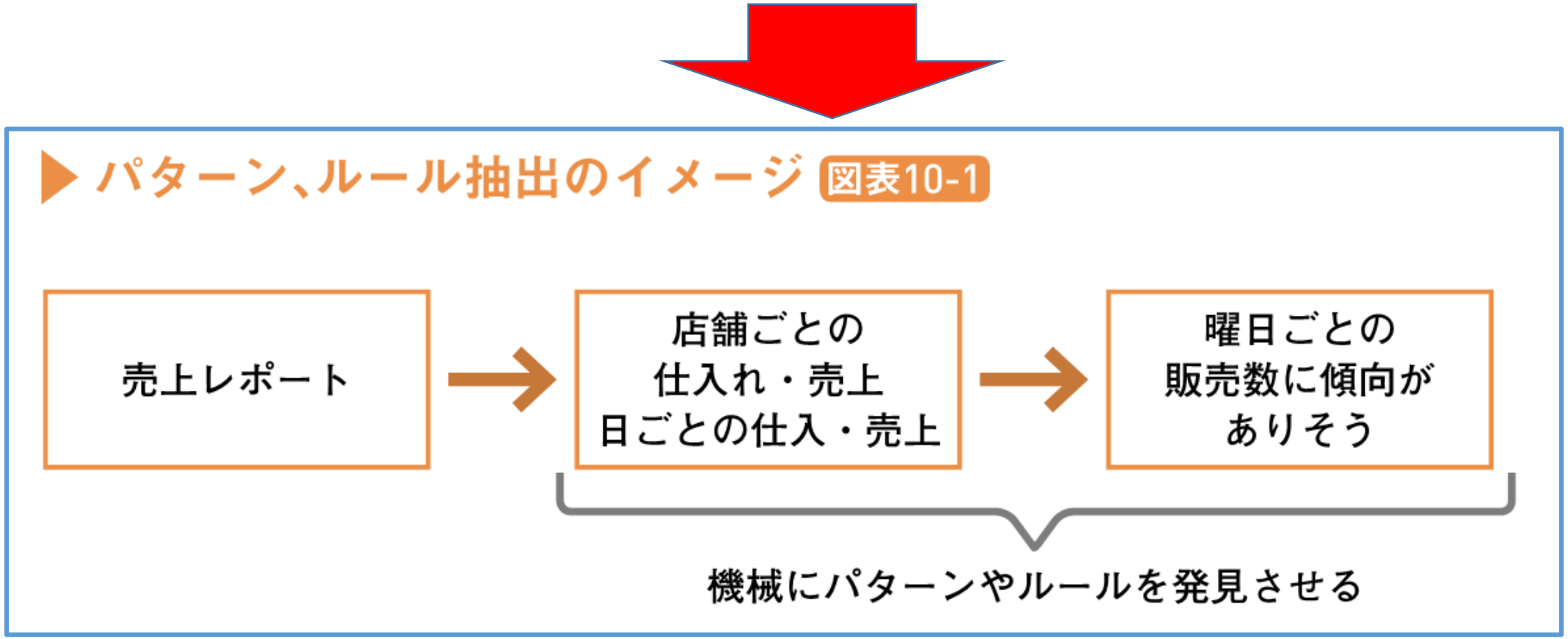
主な特徴としては、

・国内での人気が高い
・TensorFlowのように計算グラフを定義してから計算を実行するのではなく、計算と同時に計算グラフを定義するので、モデルの再構築が楽（これがPyTorchなどの動的計算グラフにつながる）

☞公式サイト：https://chainer.org/

◆機械学習の定義

機械学習 ＝ データからパターンやルールを機械自身に見つけさせる仕組み



◆機械学習の主な用途

機械学習が最も多く活用される場面

識別

判別済みのデータ（「犬」というラベルがついた画像）と未判別のデータ（ラベルがついていない画像）があるという状況で、未判別のデータを正しく識別するというタスク

予測

将来の販売数など、過去のデータにもとづいて将来どれくらい売れそうかなどの「観測されていない未来の値を予測するタスク」

◆ルールベースと機械学習の違い

ルールベース ＝ **機械学習ではなく、人が決めたルールにもとづく作業効率化の仕組み**

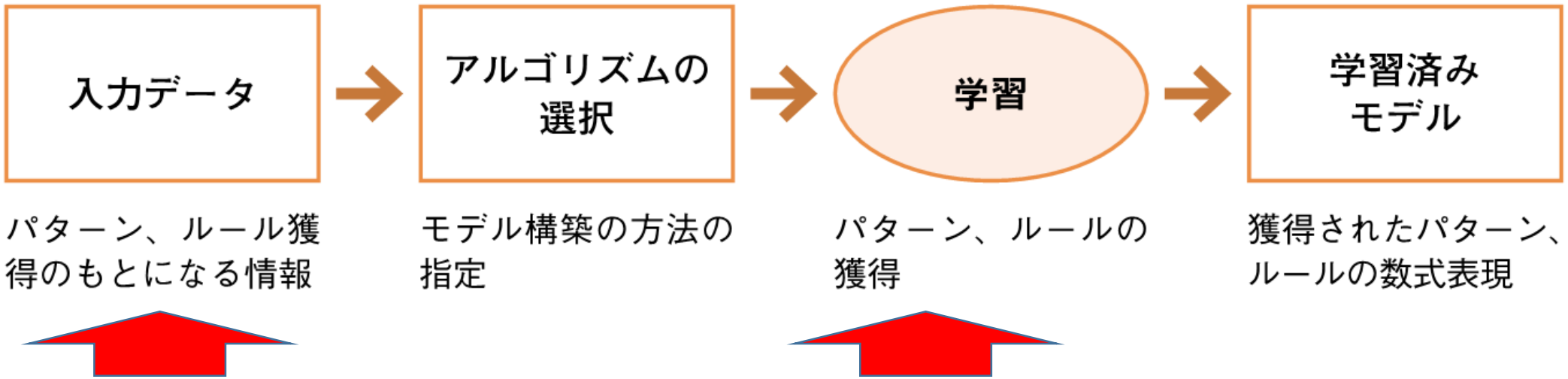
ルールベースは非常に柔軟でわかりやすく、精度をコントロールしやすい一方で、ルールが多くなると管理が難しくなるうえに、人が気づかないパターンはルールに取り込めないというデメリットもある。

▶ 機械学習とルールベースの比較 図表11-1

	機械学習	ルールベース
概要	データからルールを見つける	人がルールを決める
メリット	データの特徴に従い、アルゴリズムによって何らかの数学的根拠をもとにルールを見つける	・自明なルールを決めることが非常に簡単 ・ルールを複数組み合わせることで複雑なルールを作ることも可能
デメリット	・一定以上のデータがないと識別や予測に効果的なルールを見つけられない ・課題やデータに対して適切なモデルやアルゴリズムを選択する必要がある	・ルールのメンテナンスが大変 ・人が認識している以上の複雑で精緻なルールは作れない

◆機械学習によるパターンとルール抽出

▶ 機械学習によるモデル構築のイメージ 図表11-2



パターンやルールを発見するもとになる「データ」を用意

「機械学習アルゴリズム」を用い て「モデル」を構築する。
「モデル」：
　パターンやルールを数式で表現したもの
「機械学習アルゴリズム」：
　「モデル」を構築するための一連の数学的な処理。

注）機械学習とルールベースは組み合わせてもよい

◆機械学習が強みを発揮するにはデータが重要

データ量が少ないと信頼できる結果は得られない。たとえば、1カ月後の売上予測を行いたいのに、先週と今週のデータを見て増加傾向にあるからといって、1カ月後もこの調子で増加すると考えるのは早計

気温に応じて売れやすくなる商品があったときに、気温の情報がデータとして取得されていなければ、気温に関連するルールを見つけられない。

猫と犬の識別を行いたいのに猫のデータがなければ適切な識別は行えない。同様に、ある商品を購入しそうな人を予測したいのに、購入者のデータしか取得できない場合も正しい予測は行えない。

▶ 機械学習を行うために必要なデータの要件 図表11-3

①データの量が十分

②予測対象が適切に含まれている

③識別・予測を行うために必要な情報が含まれている

→ よい識別・予測

◆機械学習に不向きなこと

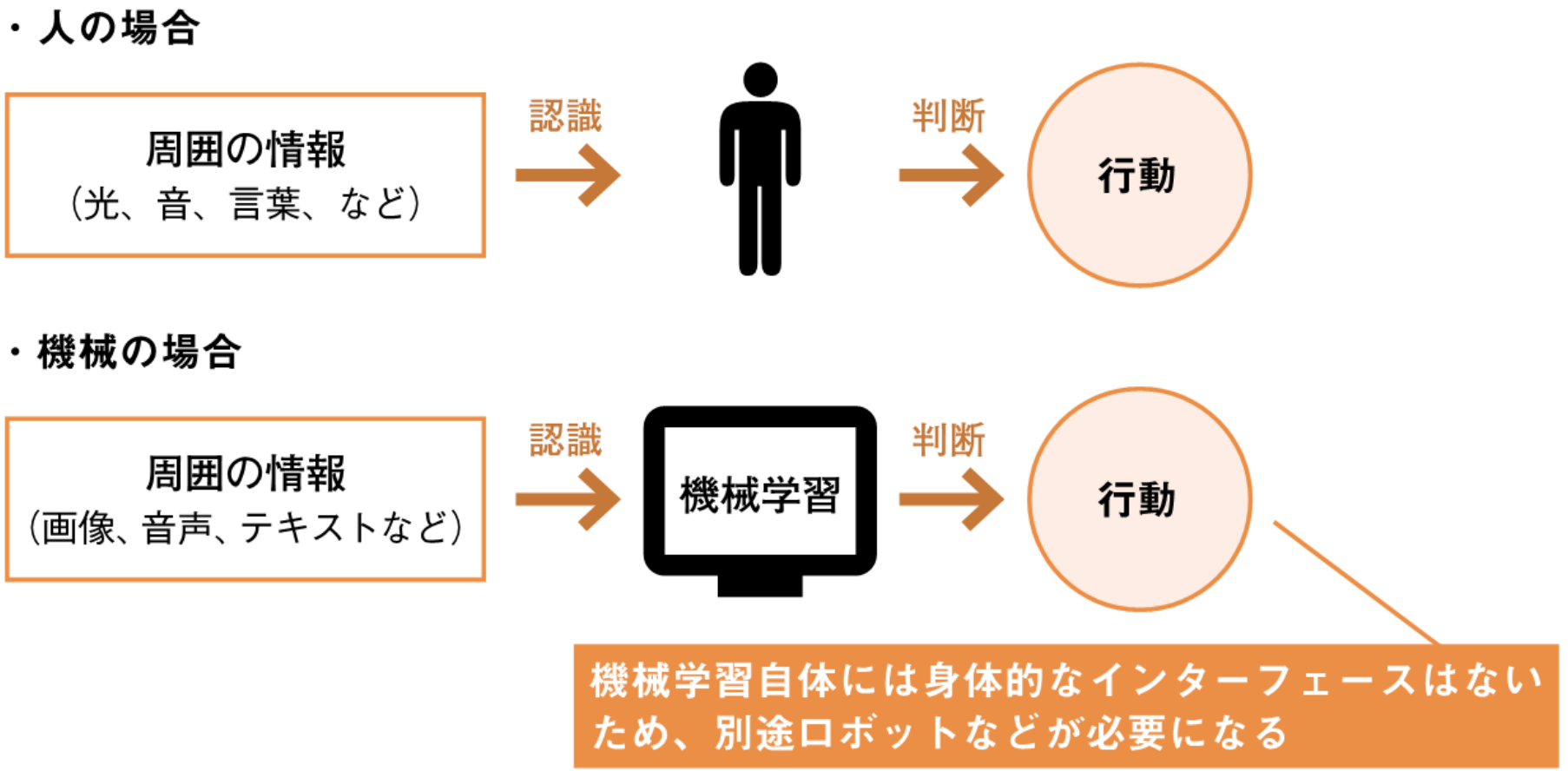
▶ 予測が困難な課題の例 図表11-4

- **偶発的に起こるもの**
  サイコロの出目、ルーレットの出目、コイントスの裏表、etc
- **現象が起こるメカニズムが複雑または現象を説明するデータが十分に取得できないもの**
  地震、メガヒット商品の発生、超高額購入顧客の発生、etc
- **過去のデータがないもの**
  新しい施策の効果、新商品の販売数、新しいの施設の売上、etc

◆機械学習の目的　①自動化によるコスト削減

機械学習は「これまで人が判断してきた作業を自動化できる可能性がある」という点が非常に優れている。

▶ 人が行う認識、判断のプロセスと機械が行う認識、判断のプロセス 図表12-1



◆機械学習の目的　②高精度の識別、予測による効率化

人による
ルールの識別

◎　個別の要因からルールを識別し、将来を予測する

×　膨大なデータから、それぞれのデータの関係を読み取って
ルールを識別し、将来を予測する

機械学習は、良いデータと、良いアルゴリズムを選び、
うまく学習させると非常に高い性能を発揮する

DeepMind社の機械学習技術による、「データセンター内の冷 却電力の効率化」。
データセンター内外のさまざまな場所に取りつけたセンサーのデータを学習することで、設備の稼働状況や周辺の環境の変化に応じた冷却設備の稼働コントロールを実現。
結果、従来と比べて約40%の電力の削減が図られたと発表されている。

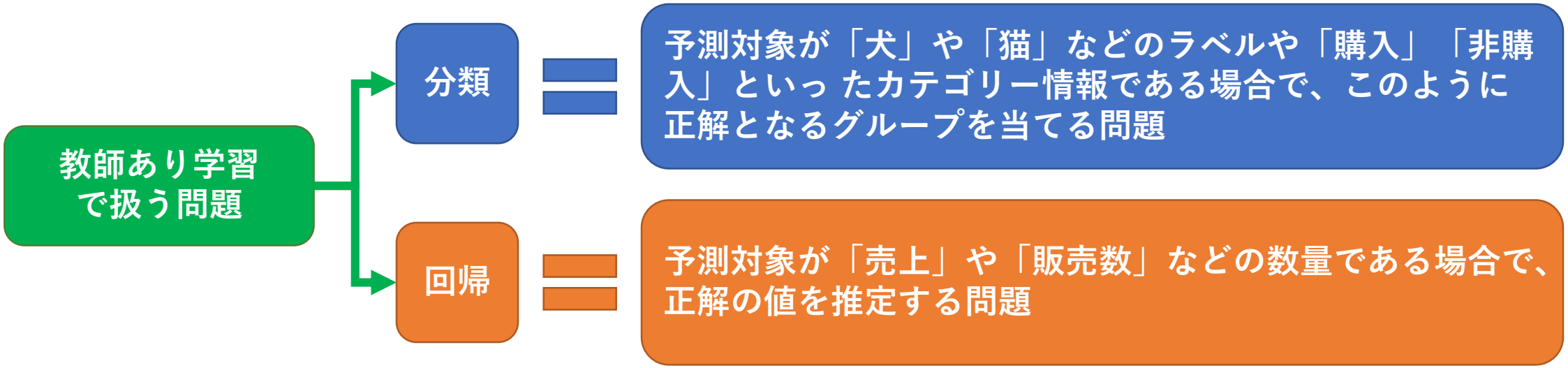
◆機械学習の３つの手法

①「教師あり学習」
入力に対してあらかじめ正解がわかっている場合に、正解を導くパターンやルールを学習する手法。
ここでの「教師」とは、正解データのこと。
②「教師なし学習」
正解のないデータから類似グループをまとめたり、重要な特徴を抽出したりする学習手法。
③「強化学習」
コンピューターが自ら試行錯誤しながら最適な戦略を学習する手法。

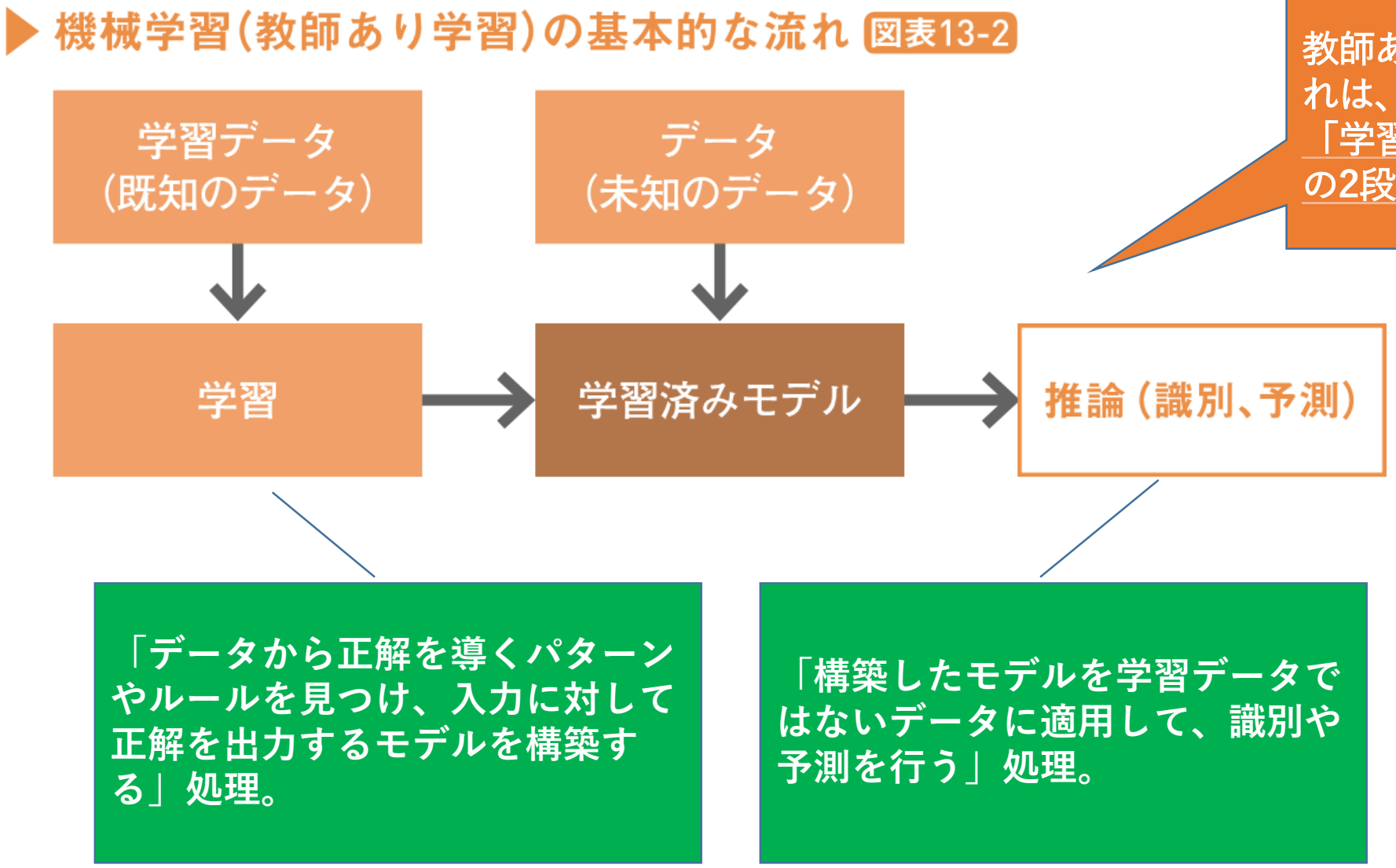
▶ 機械学習の3つの手法 図表13-1

手法	機能
教師あり学習	正解のラベルや数値がわかっているデータをもとに学習モデルを構築し、ラベルや数値が未知のデータに対して予測や識別を行う手法。正解データがない場合は人がデータに正解をつける必要がある。たとえば大量のペットの画像データがあったとして、それぞれに「犬」や「猫」といったラベルをつけて学習させ、学習モデルを作る。以後は、ラベルがついていないデータを学習モデルに与えると、犬の画像なら犬、猫の画像なら猫に自動的にラベルづけされる
教師なし学習	正解のないデータから、共通する特徴を持つグループを見つけたり、データを特徴づける情報を抽出したりする学習手法。たとえば、購買データから購買行動が似ているユーザーをグルーピングしたり、アンケートデータからユーザーの嗜好（ブランド好き、アウトドア好きなど）を抽出したりすることができる
強化学習	ゲームやギャンブルなど、結果がでるまでに時間がかかったり、多数の繰り返しが必要になったりするタスクに関して、実際に行動しながら最適な戦略を学習する手法。たとえばゲームなどで、コンピューター自身がどのように行動したら高得点が得られるかを試行錯誤しながら学んでいく

◆教師あり学習でできること



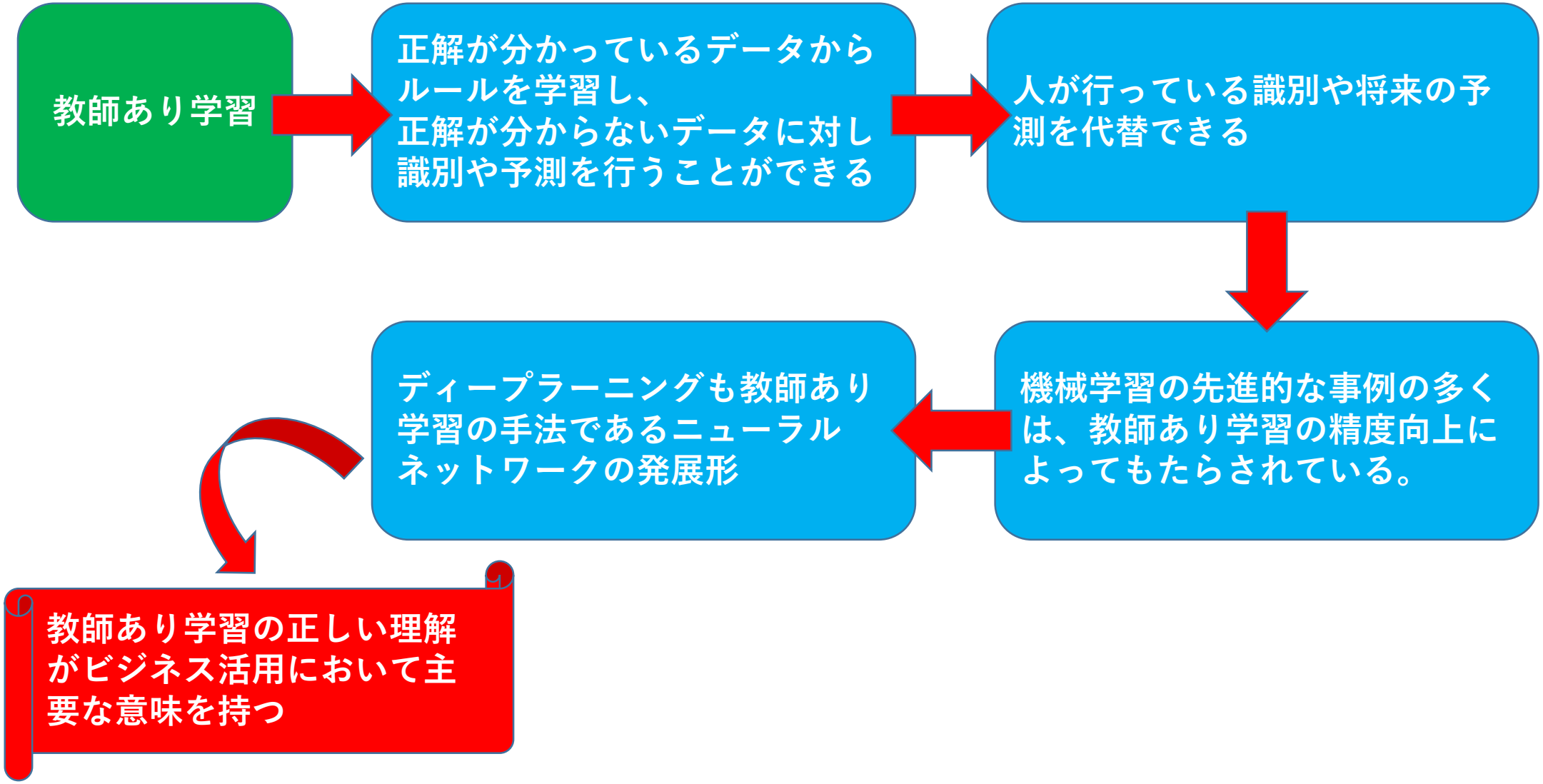
◆機械学習（教師あり学習）の基本的な流れ



教師あり学習の一般的な流れは、大きく分けると「学習」と「推論」の2段階で構成されている。

適切な学習により構築されたモデルは、推論においてもよい精度を持つようになる。
そのため、よい推論が行えるようなモデルを構築することが機械学習における主な目標になる。

◆教師あり学習の特徴


教師あり学習
正解が分かっているデータからルールを学習し、正解が分からないデータに対し識別や予測を行うことができる
人が行っている識別や将来の予測を代替できる
機械学習の先進的な事例の多くは、教師あり学習の精度向上によってもたらされている。
ディープラーニングも教師あり学習の手法であるニューラルネットワークの発展形
教師あり学習の正しい理解がビジネス活用において主要な意味を持つ

◆教師なし学習とは？

教師なし学習

識別や予測の対象となる正解（教師）がない学習の手法

類似データのグルーピング
👇
クラスタリング

クラスタリングとはデータ間の距離に基づいて類似データをグルーピングすること

「乗用車ブランドの好意度を尋ねたアンケート」があるとして、車に対するイメージ（高級車好き、性能重視、安価志向、安全性重視など）が近いユーザーをまとめるといったことができる。

各グループの嗜好に合った内容のダイレクトメールを送る

次元削減

次元削減とはデータの次元（特徴量の数）を削減する手法

次元削減を行うことで、より本質的なデータを抽出することが可能となる

「主成分分析」と呼ばれる手法で、少ない情報で全体のばらつきを表現する主成分という特徴を抽出することができる

商品イメージに関するアンケートから、複数の評価項目をまとめて「安全・安心」「価格」「機能性」「話題性」に関するイメージのような少数の要素を抽出するといったことが可能

◆強化学習とは

強化学習

試行錯誤を繰り返しながら学習を進めていく手法

強化学習は、教師あり学習や教師なし学習とはそもそもの問題設定が異なる

強化学習を理解するには、
「環境」
「行動」
「報酬」
という3つの概念が重要。

環境	ある時点の状態
行動	その状態での実際の行動
報酬	その行動によって上がったスコアや勝利に対する貢献

主にゲームやギャンブルなど、結果が出るまでに時間がかかるタスクや多数の繰り返しが必要になるタスクに関して、コンピューターが実際に行動しながら最適な戦略を学習していく。

AlphaGoはまさに強化学習が力を発揮した事例。
AlphaGo同士がコンピューター上で対局を繰り返しながら強くなっていき、人間を超える強さを獲得している。

◆機械学習のモデルとアルゴリズム

教師あり学習におけるモデルとアルゴリズム

教師あり学習は推定や予測をする問題に適用できる

推定や予測
☞
回帰

機械学習では識別や予測を行うために、モデルを学習アルゴリズムにもとづいて構築する。

モデルとは学習アルゴリズムによって獲得されたパターンやルールを数式的に表現したもの

ある商品の売上を気温と湿度から予測する場合の線形回帰モデルの例

▶ 線形回帰の例 図表14-1

$$\text{予測売上} = \alpha \times \text{気温} + b \times \text{湿度} + \text{定数}$$

- 目的変数：予測売上
- 重みパラメーター：α、b
- 説明変数：気温、湿度
- 定数：気温と湿度が0の場合の売上に相当

売上が気温と湿度に応じて変動する場合の線形回帰による売上予測の場合。重みパラメーターと定数部分を「最小二乗法」というアルゴリズムによって求める

説明変数：識別や予測を行う対象となる変数

目的変数：識別や予測を行うために必要となる変数

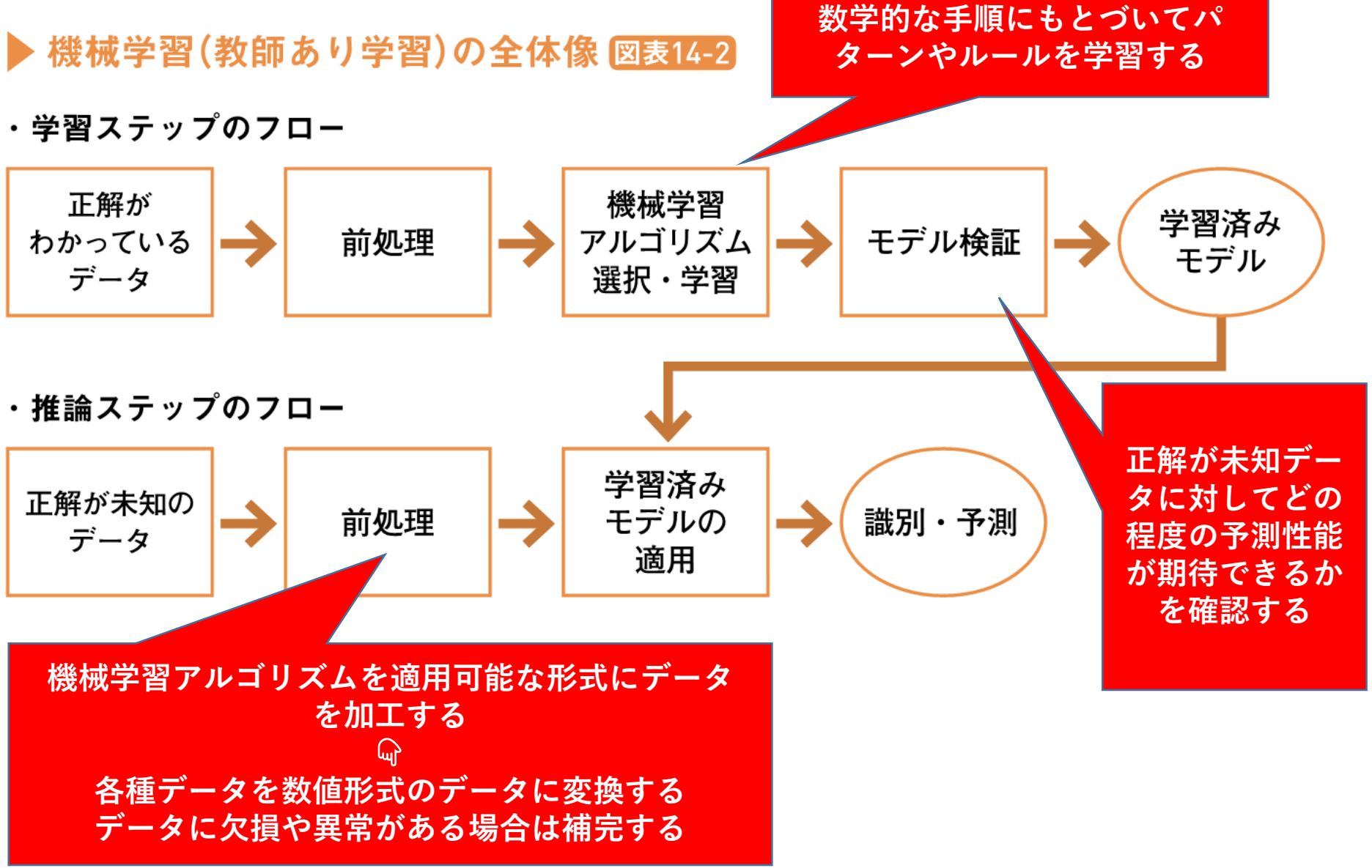
◆モデル構築のフロー

顧客データ売上データ
などの表形式のデータ

画像
JPEG、GIF
など

音声
MP3、MP４

自然言語
テキスト



◆データの種類と前処理・構造化データと非構造化データ

機械学習アルゴリズムを用いてモデルを構築するためには、データの形式を適切に変換する「前処理」が必要

機械学習で使用するデータ

構造化データ

CSVデータやExcelデータ

非構造化データ

ディープラーニングが発展したことにより、画像やテキスト、音声データに関する識別性能が大幅に向上

「画像」「テキスト」「音声」といったデータ

▶ 構造化データ・非構造化データの例 図表15-1

構造化データ	非構造化データ
・**各種業務システム内のデータ**（受注、発注、在庫、人事、POS など） ・**政府や調査会社の統計データ**	・**画像データ**（商品画像、SNS の投稿画像など） ・**動画データ**（監視カメラ映像、テレビ番組など） ・**テキストデータ**（議事録、SNS 投稿テキストなど） ・**音声データ**（コールセンター会話録、会議録音データなど）

◆構造化データに対する機械学習

顧客ごとのある商品の
購入確率を予測したい

顧客ごとの商品購入経験の有無と、商品購入に影響を与えると考えられる情報を行列形式に整理する

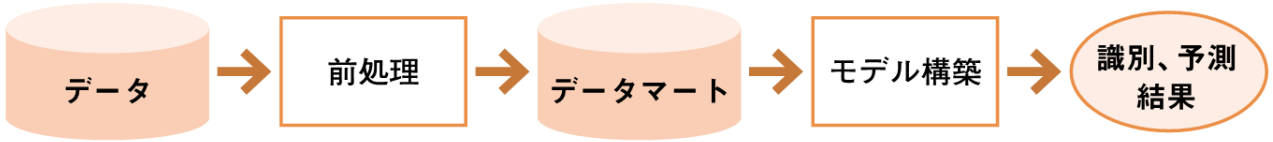
データが欠落していたり、異常な値のデータが含まれている場合、適切にデータを補完したり、異常データを除去する

前処理

データマート

事例

▶ 構造化データにおけるモデル構築のイメージ 図表15-2



- POS データ
- 会員データ
- 店舗データ
  etc

- データの結合
- 集約処理
- 欠損値処理
- 異常値除去
  etc

モデル構築に必要な情報で作られるテーブル

- 線形回帰
- ナイーブベイズ
- 決定木
- ランダムフォレスト
  etc

顧客ID	年齢	性別(男性:1、女性:0)	来店回数	最終来店後経過日数	…	商品A購入
1001	40	1	11	35	…	1
1002	29	0	5	12	…	0
[illegible]	48	1	[illegible]	50	…	[illegible]
1009	[illegible]	[illegible]	3	[illegible]	[illegible]	0
1010	18	0	12	10	…	1
…	…	…	…	…	…	…

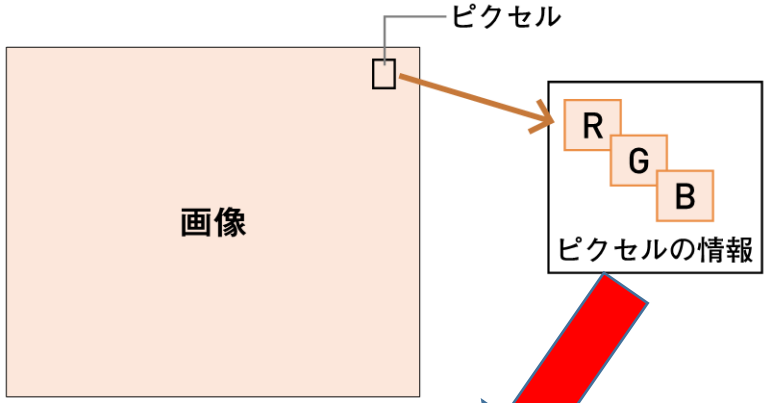
説明変数
（パターン、ルールを見つけるもとのデータ）

目的変数
（識別、予測の対象データ）

特徴量

◆画像データに対する機械学習と画像認識のタスクの種別

▶ 画像データの構造 図表15-3



各ピクセルの情報を入力データとして、機械学習のアルゴリズムに適用

画像認識：与えられた画像に写っているものを識別するタスク

- **①画像分類**：画像に写っている物体を特定のカテゴリーに分類するタスク
  - 犬が写っている画像に「犬」というラベルを付与するという処理
- **②物体検出**：特定のカテゴリーに属する物体が画像内のどこに写っているかを検出するタスク
  - 人が写っている画像から「顔」にあたる部分のみを抽出するような処理
- **③シーン理解**：画像がどのような状況を意味しているかを識別するタスク
  - 横断歩道と人が写っている画像から、「人が横断歩道を渡ろうとしている」という状況を識別する処理

◆自然言語データに対する機械学習

自然言語に機械学習を適用する

文章を数値的なデータとして表現する

代表的な方法 Bag of Words 表現

文章ごとに単語の出現数をカウントし、文章と対応する単語の要素を列として表形式で表現

日本語の文章に Bag of Words を使用するとき

形態素解析 ☞「品詞によって分割」を使用して分割する

自然言語データにおける前処理

▶ Bag of Words表現 図表15-4

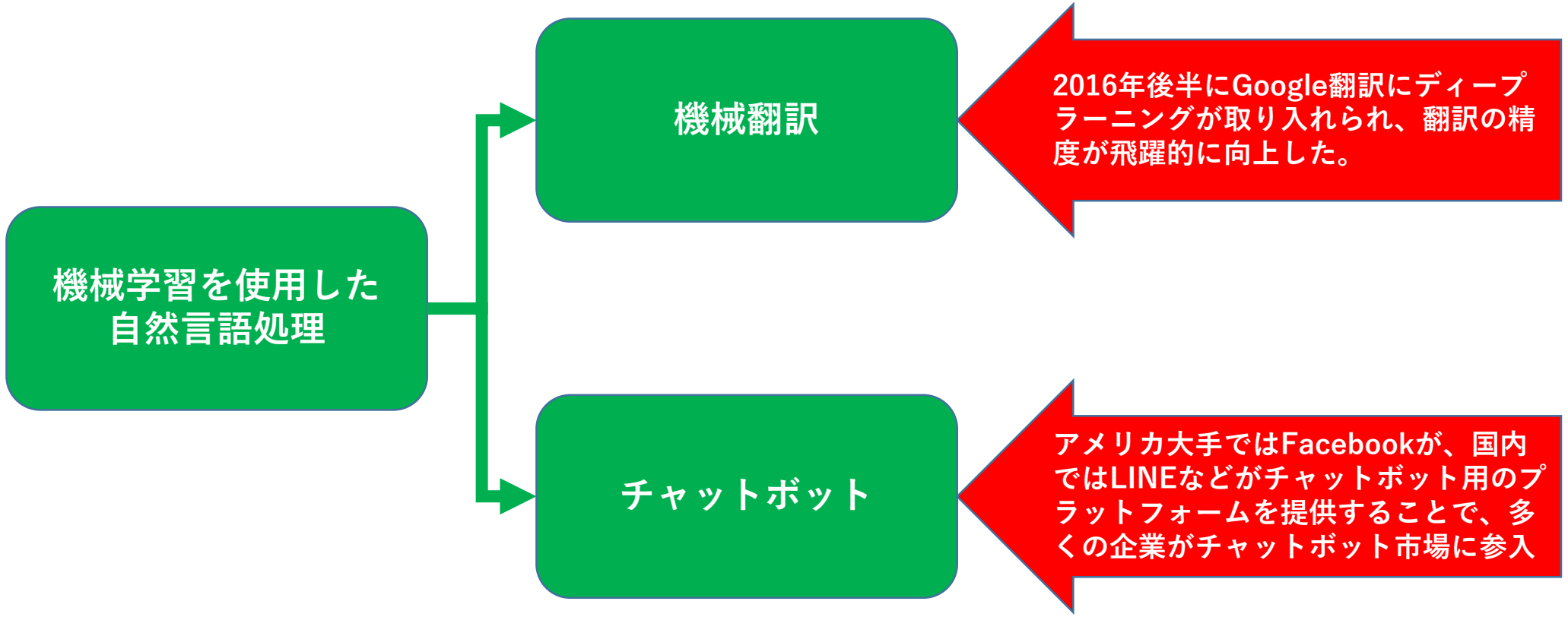
元の文章	形態素解析による分割
これはペンです	これ／は／ペン／です
あれは車です	
それは鉛筆です	

「これはペンです」という文章をBag of Wordsによりベクトル化した例。文章に出現する「これ」「ペン」「です」に数字の「1」が入り、ほかの文章に含まれる単語はカウントされず「0」になる。このようにして数値の羅列に変換することをベクトル化と呼ぶ

ベクトル化

これ	あれ	それ	…	鉛筆	ペン	車	…	です	ない
1	0	0		0	1	0		1	0

◆自然言語処理の活用事例



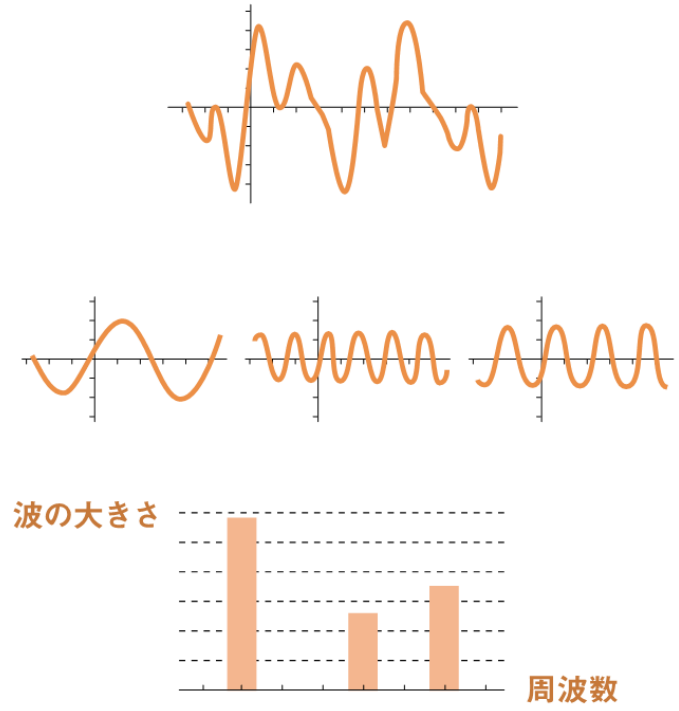
◆音声データに対する機械学習

音声データ → さまざまな波形が入り混じった時系列のデータ → フーリエ変換という手法を用いて周波数ごとの特徴に変換する → 周波数別の情報を特徴量として機械学習を用いることが可能

音声データの活用例

- 音声アシスト機能
  - 発話された音声を認識し、内容を理解したうえで発話者に対して望ましい回答を返す
- コールセンター業務への適用

▶フーリエ変換のイメージ 図表15-5



複雑な波形の元データ

↓

波の大きさと周波数に応じて分解する

↓

波の大きさと周波数で表す

◆学習アルゴリズムの選択

これまでにさまざまな学習アルゴリズムが考案されている

- 線形回帰
- ランダムフォレスト
- ナイーブベイズ
- ニューラルネットワーク
- 決定木
- サポートベクターマシン

それぞれ適用可能な状況やパフォーマンスを発揮するための前提条件が異なる

目的や状況に応じて適切な手法を選ばなくてはならない

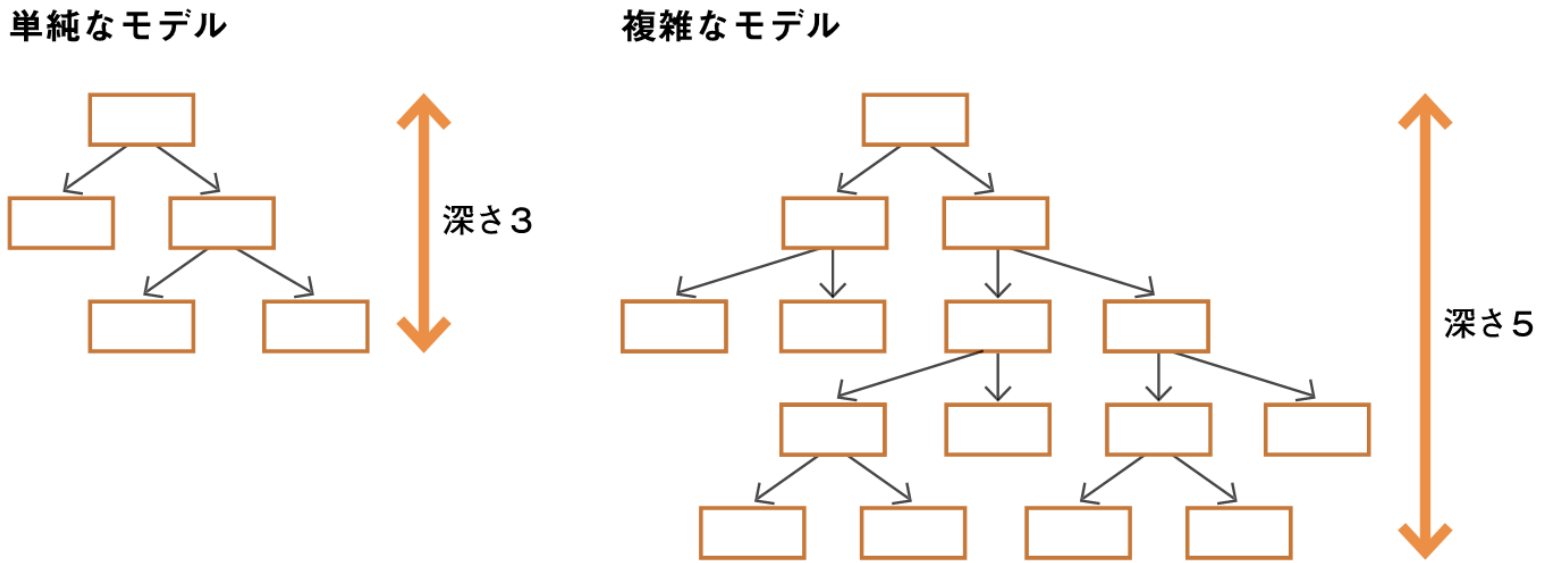
▶ 教師あり学習の主なアルゴリズム 図表16-1

手法	説明
線形回帰 (Linear Regression)	目的変数と説明変数に線形関係を仮定してモデルを構築する手法。予測値に対する各特徴量の寄与度がわかるため、構造の解釈がしやすい
ナイーブベイズ (Naive Bayes)	特徴量間に独立性を仮定して、各カテゴリに所属する確率を推定する手法。主に文書分類などに利用される
決定木 (Decision Tree)	木のように段階的にルールを適用しながらデータを分類していく手法。ルールを辿っていくプロセスが可視化されるため、構造の解釈がしやすい
ランダムフォレスト (Random Forest)	事例と特徴量をランダムにサンプリングしたデータをもとに決定木を多数作り、多数決的に回帰・分類を行う手法
ニューラルネットワーク (Neural Network)	人間の脳の活動を模した学習の手法。ニューロンと呼ばれる層をつなぎ合わせてネットワークを構築する
サポートベクターマシン (Support Vector Machine)	カーネル関数といわれる高次元空間への写像関数を用いて分類を行う手法。線形分離不可能なデータも分類できる一方で、計算コストが高い

◆モデルの複雑さを調整するハイパーパラメータ

ハイパーパラメータ → モデルの複雑さを調節するパラメータ → 決定木においては木の深さ → ニューラルネットワークにおいてはユニットや層の数

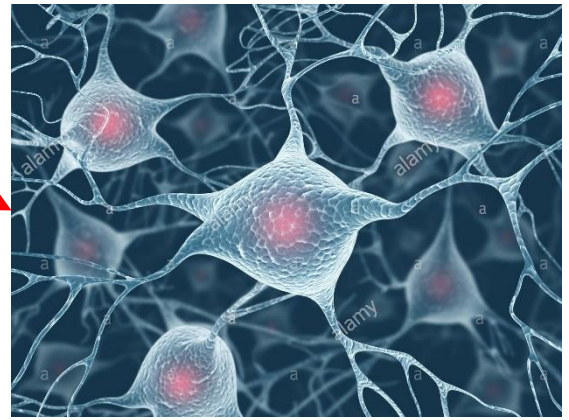
▶ 決定木におけるハイパーパラメーターの例 図表16-2

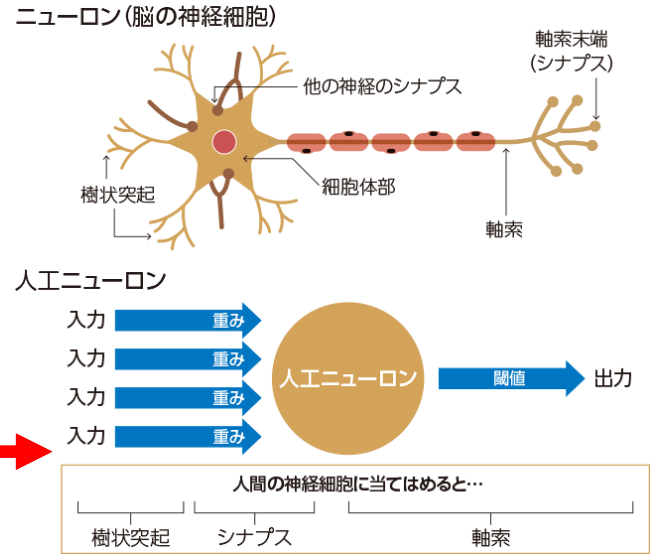


木の深さがハイパーパラメーターに該当し、深いほうが複雑なモデルを意味する

◆ニューラルネットワークとディープラーニング

人の脳はニューロンと呼ばれる多数の神経細胞で構成されており、ニューロンが複雑に結合することによって情報処理が行われていると考えられている





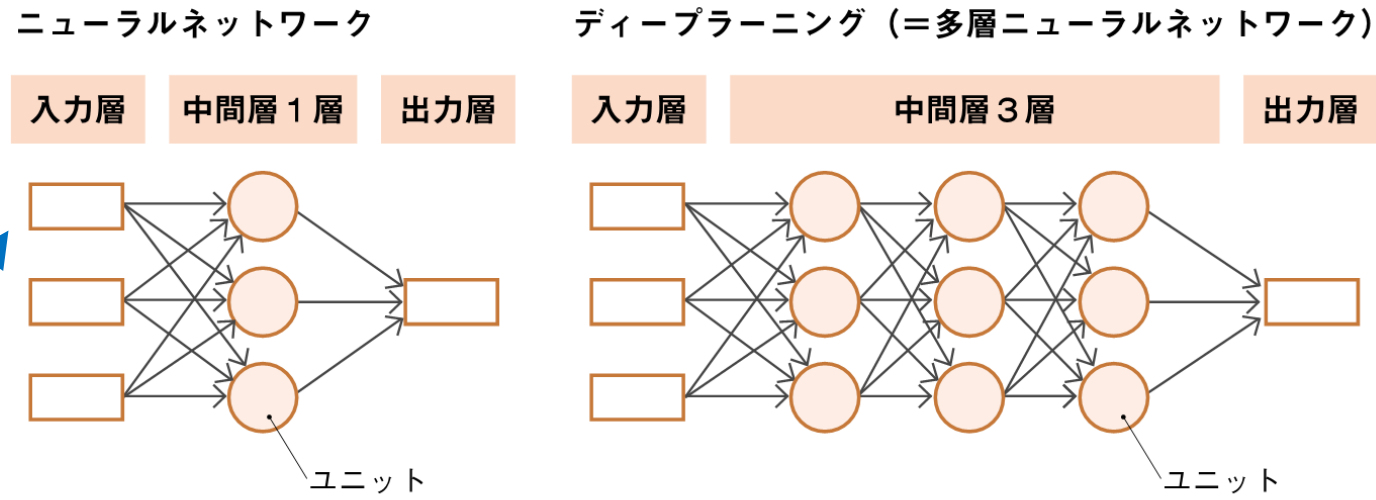
この情報処理ネットワークを模したモデルがニューラルネットワークである

ニューラルネットワークはニューロンに該当する多数のユニットから構成される

各ユニットは入力に対して、ユニット間の重みを加味した非線形な演算を行った結果を出力するという構造を持つ

このときニューラルネットワークの層を多数重ねたものを「ディープニューラルネットワーク」と呼び、ディープニューラルネットワークを用いた機械学習の手法を「ディープラーニング」と呼びます

▶ ニューラルネットワークとディープラーニングのイメージ 図表17-1



ニューラルネットワークの中間層を多層構造にしたモデルをディープラーニングと呼ぶ

◆ディープラーニングの学習について

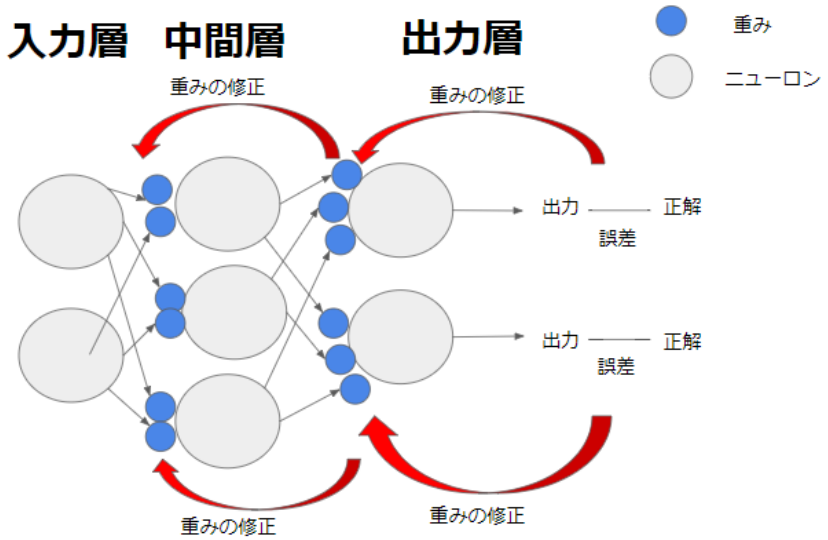
ディープラーニングの学習

「誤差逆伝搬法」（バックプロパゲーション）アルゴリズムを用いる

正解との誤差が段々小さくなるようにユニット間の重みを調整していくことで実現される

層が増えると誤差逆伝搬法が機能しなくなる問題が起きるため、適切に学習できないということがあった

アルゴリズムの改善と計算資源の向上により、近年では多くの問題に対してほかのモデルが追随できない精度を達成するようになってきた



▶ さまざまなディープラーニング関連のモデル 図表17-2

手法	説明
ディープニューラルネットワーク（Deep Neural Network：DNN）	ニューラルネットワークの層を多層にしたモデル。一般的には中間層が2層以上の深い構造を持つ
オートエンコーダ（AutoEncoder）	ニューラルネットワークの出力が入力と一致するように構成したモデルであり、主に次元削減のために利用される
畳み込みニューラルネットワーク（Convolutional Neural Network：CNN）	主に画像解析に用いられるモデルで、局所的な領域の情報を畳み込む処理を行う畳み込み層と特徴を集約するプーリング層を多層に組み合わせて構成される
リカレントニューラルネットワーク（Recurrent Neural Network：RNN）	主にテキスト解析や時系列解析に用いられるモデルで、隠れ層の値を再び次の層の入力として用いるという再帰的な構造を持つ