

令和2年度「専修学校による地域産業中核的人材養成事業」

# 人工知能教育研修 プログラム教材

令和2年度文部科学省委託  
「専修学校による地域産業中核的人材養成事業」事業

# 人工知能教育研修プログラム教材

教員用テキスト

Society5.0実現のためのIT技術者養成モデルカリキュラム開発と実証事業

学校法人電子学園

## ■スケジュール

研修テーマ	研修内容	時間
第1章 人工知能とは		
1-1. 人工知能の歴史	人工知能の歴史はブームと停滞期の連続であることを理解する	1時間
1-2. 人工知能とは	ルールベース、機械学習、人工ニューラルネットワーク、ディープラーニング、それぞれの学習概念を理解する	
1-3. 機械学習		
1-4. 人工ニューラルネットワーク		
1-5. ディープラーニング（深層学習）		
第2章 機械学習のタイプ		
2-1. 機械学習とは	機械学習における基本的な目標とタイプ、教師あり学習と教師なし学習の違いについて理解する	1時間
2-2. 機械学習のタイプ（分類）		
2-3. 教師あり学習		
2-4. 教師なし学習		
2-5. 教師あり学習と教師なし学習のアルゴリズム	機械学習における基本的なアルゴリズムと強化学習の概念を理解する	1時間
2-6. 強化学習	機械学習におけるデータの重要性と注意すべき点、入手方法について理解する	1時間
2-7. 機械学習のためのデータ		
2-8. 学習データで注意すべき点		
2-9. データの入手	教師あり学習の学習フェーズ、評価・応用フェーズの流れを理解する	1時間
2-10. 教師あり学習の流れ		
第3章 ディープラーニング		
3-1. ディープラーニング（深層学習）とは	ディープラーニングにおける画像と映像の認識の概念、人工ニューロンの概論について理解する	1時間
3-2. 視覚情報の重要性		
3-3. 生物学ニューロンから人工ニューロンへ		
3-4. ニューラルネットワーク	多層パーセプトロンのニューラルネットワークの基本的な構造、ディープラーニングの応用領域について理解する	1時間
3-5. 実用化が進むディープラーニング		
3-6. ディープラーニングに必要な数学		

研修テーマ	研修内容	時間
第4章 画像処理の基礎		
4-1. 人工知能による画像認識処理	人工知能による画像処理の概念と画像認識の流れを理解する	1時間
4-2. 画像認識とは		
4-3. 文字の画像認識のステップ		
4-4. 人工知能による画像認識の方式		
4-5. ニューラルネットワークによる認識	ルールベース、機械学習、ディープラーニング、それぞれの画像処理について理解する	1時間
4-6. ディープラーニングによる認識		
4-7. 人工知能を使った画像処理		
第5章 Pythonプログラミング		
5-1. Pythonとは	Pythonとは何か、どのような開発に活用されているのかを理解する	1時間
5-2. 数多くのWebアプリケーションの開発に活用		
5-3. 人工知能の開発に活用		
5-4. Pythonプログラムの基本事項	Pythonプログラムの実行方法と保存・読み出し方法を理解する	1時間
第6章 データの前処理技術		
6-1. 人工知能の実装プロセス	人工知能の実装のプロセスよりデータ前処理の重要性とデータ前処理で行うことの意味を理解する	1時間
6-2. 機械学習に向けてのデータ前処理		
6-3. データの前処理はなぜ必要か		
6-4. データの前処理		
6-5. 機械学習のためのデータ前処理	機械学習のためのデータ前処理とその基本的なオペレーションについて理解する	1時間
6-6. 前処理オペレーション		

## 第1章 人工知能とは

### 1-1. 人工知能の歴史

# 人工知能 (AI) = Artificial Intelligence

人工知能はこの数年で突然発生した学問ではありません。長い歴史を持った学問です。

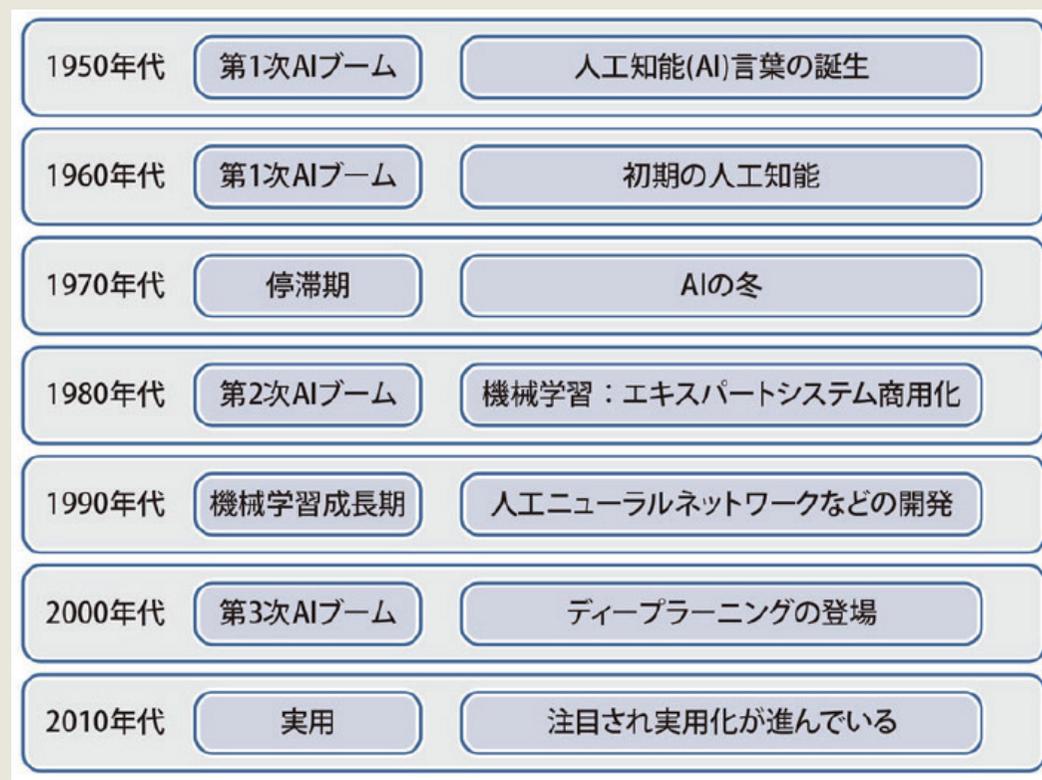
機械翻訳のための**自然言語処理**、自動運転を実現するための**画像認識**、音声入力のための**音声認識**といった技術は、人工知能の研究から生まれたものです。人工知能は、その時間軸上も最初に登場した言葉であり、概念的にも一番広い意味を持つ言葉になります。

人工知能という言葉の誕生は1950年代にまで遡ります。その次に第1次AIブームが訪れて、停滞期とブームが繰り返してきています。長い時を経て2000年に入り、第3次AIブームに突入し、今日のディープラーニングの手法が確立されました。

IBMのAI「ワトソン」がクイズ番組で優勝し、Alpha Goはトップ棋士に勝利し、Google傘下のDeepMind社が開発したAlphaZeroがルールの強化学習のみで、チェス、将棋、囲碁のそれぞれの世界最強AIを打ち負かすなど、AIは既に驚異的なレベルにきています。

## 第1章 人工知能とは

図1 人工知能の歴史



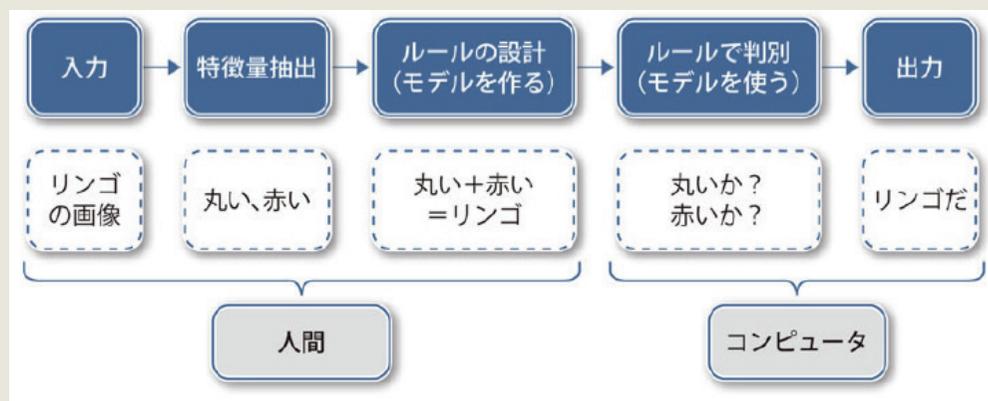
## 第1章 人工知能とは

### 1-2. 人工知能とは

初期の代表的な人工知能の実装はルールベースでした。図2のように、リンゴの画像に対して、「丸い」、「赤い」という特徴を把握すれば、リンゴとして判別して認識は成功します。しかし、緑のリンゴではこの認識は成功しません。そのため、「丸い、赤いあるいは緑」といったようにルールを追加していきます。

しかし、すべてのパターンを網羅的にカバーするのは限界があります。ルールベースで実装されているシステムは、この限界を超えることはできないのです。

図2 従来のルールベース（人工知能の学習概念図）



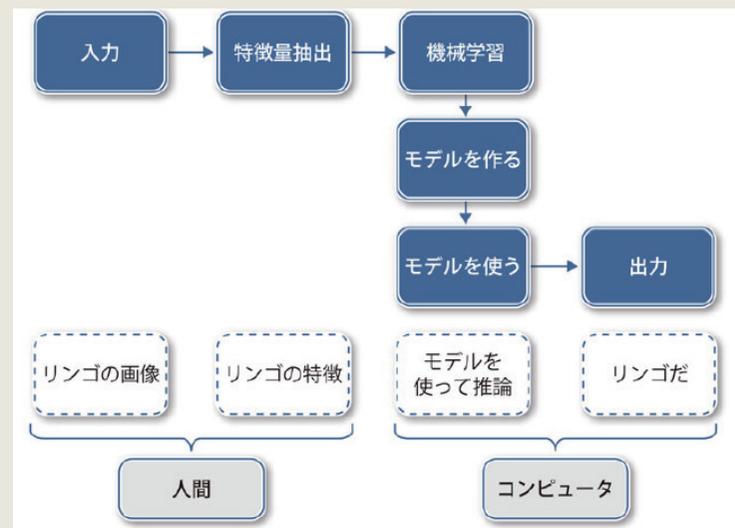
# 第1章 人工知能とは

## 1-3. 機械学習

一般的に機械学習のモデルを構築するためには、膨大な量のデータを学習させ、そのデータ内の傾向を取得する必要があります。データを学習させるために、与えられたデータの特徴が数値化されている必要があります。この対象の特徴が数値化されたものを**特徴量**と言います。

リンゴの認識にあたり、大量のリンゴの特徴を用紙します。色、重さ、体積といった特徴を数値化してコンピュータに機械学習させ、それらの関連性を見つけ出せるようにします。

図3 機械学習の概念図



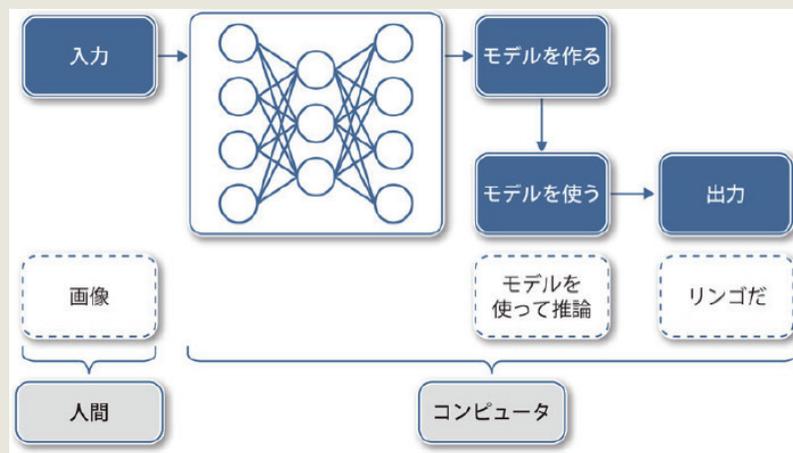


## 1-5. ディープラーニング（深層学習）

ディープラーニング（深層学習）は、学習の原理は人工ニューラルネットワークと同じですが、**人工ニューラルネットワークの層を複数重ねること**によって、学習の精度を上げる試みになります。ディープラーニングは、人工ニューラルネットワークの階層数を増やすことで、より複雑な計算と膨大な計算量になりますが、よりたくさんの「特徴」を見つけることができます。

この段階にまで来ると、人間が用意するものはたったの二つです。一つは学習させたい大量のデータ、もう一つはパワフルな計算機器です。

図5 ディープラーニングの概念図



## 第2章 機械学習のタイプ

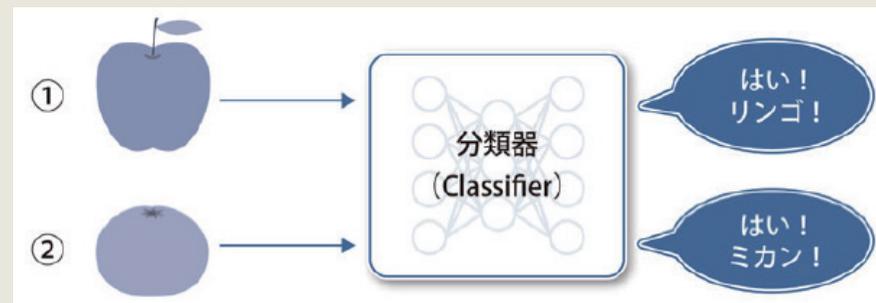
### 2-1. 機械学習とは

# 機械学習 (ML) = Machine Learning

機械学習とは、人間が明示的にプログラミングせずに、コンピュータがデータから学習できるためのプログラミングの手法です。

抽象的な表現ではありますが、機械学習においては**分類器 (Classifier)** を作ることを目標の一つです。図6のように、リンゴとミカンの写真を見せたら、それがリンゴからミカンかを「分類」してくれるものです。機械学習では、この「分類器」を作るのは人間ではなく、学習でコンピュータが作ります。

図6 ①分類器・リンゴとして認識、②分類器・ミカンとして認識



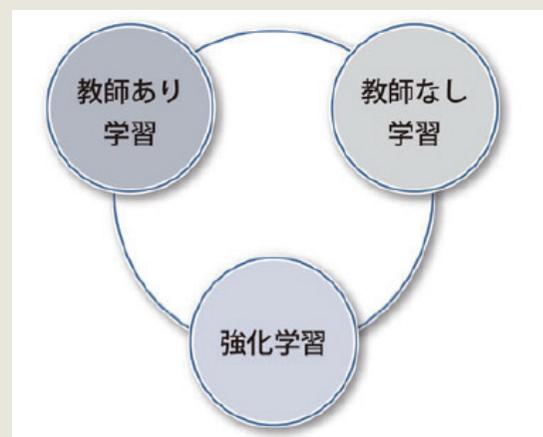
## 第2章 機械学習のタイプ

### 2-2. 機械学習のタイプ（分類）

人工知能の研究は長い間、ルールベースのアプローチを取ってきました。ルールを設定するという一番重要な役割は人間が担当していたのです。コンピュータ分類器を作れるようになったことで機械学習では、人間が担当しているところを可能な限りコンピュータが実行できるようになりました。

機械学習のタイプは大きく3つあります（図7）。ほかにも、バッチ学習、オンライン学習、モデルベース学習などさまざまな機械学習タイプがあります。

図7 機械学習の分類図解



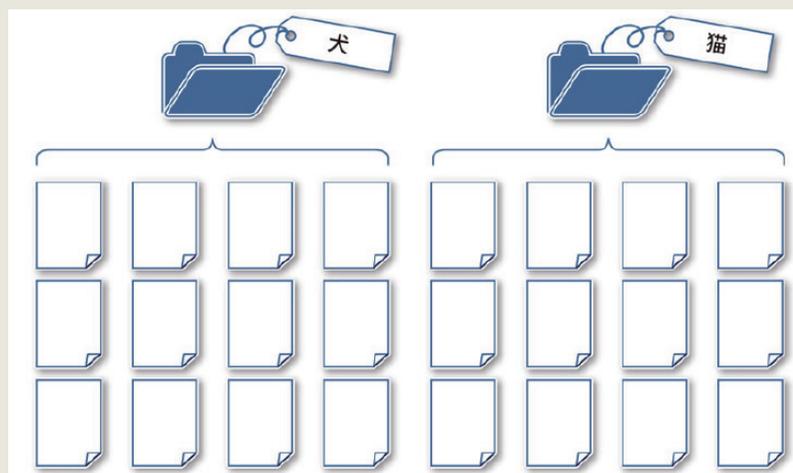
## 第2章 機械学習のタイプ

### 2-3. 教師あり学習

機械学習でいう「**教師**」は、データに付随している正解となるラベルのことです。例えば、犬と猫の写真のデータであれば、その写真に写っているのは猫なのか、犬なのかというラベルです。

図8のように、便宜上同じラベルの写真データを同じフォルダに入れて、そのフォルダ名がラベルになっているケースもあります

図8 フォルダがラベル（教師）になる

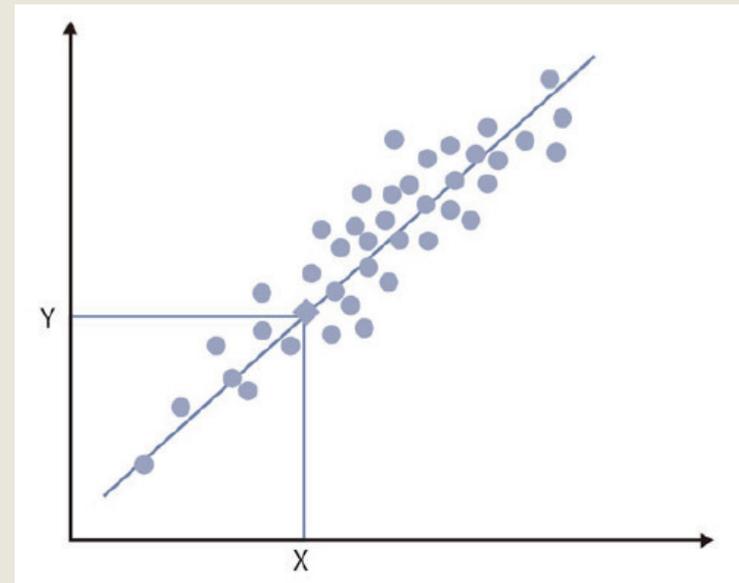


## 第2章 機械学習のタイプ

### ■教師あり学習の用途

分類 classification	機械学習によって学んだ結果、メール受信後すぐに迷惑メールフォルダに振り分けます。この迷惑メールフィルタが「分類」という処理です。例えば、リンゴは「0」、ミカンは「1」というクラスに分けるという具合です。
回帰 regression	分析対象の一連の <b>特徴量</b> からターゲットの数値を予測することを「回帰」といいます。例えば、中古アパートの家賃を予測するとき、築年数・立地・家賃等が特徴量となり、ターゲットは家賃になります。

図9 縦型回帰の例



対象データとなる特徴量の分布から、そのデータを「表現」できる直線を見つけることで、予測したいデータを計算によって導くことができるようになります。

## 第2章 機械学習のタイプ

### 2-4. 教師なし学習

教師なし学習は、正解となるラベルがないものになります。例えば、写真は大量にありますが、写っているのは犬なのか猫なのかというラベルはありません。その状態から規則性とパターンを発見するのが教師なし学習の目的となり、相関ルール学習や異常検知も含まれます。

#### ■教師なし学習の用途

クラスタリング clustering	クラスタリングは、データの属性によってグループになる傾向を見つけ出すことです
アソシエーション分析 Association Analytics	アソシエーション分析では、データの大部分を表すようなルールを見つけ出します。 (Xを買う人はYも買う傾向にあるなど)

図10 データのクラスタリング（グループ分け）

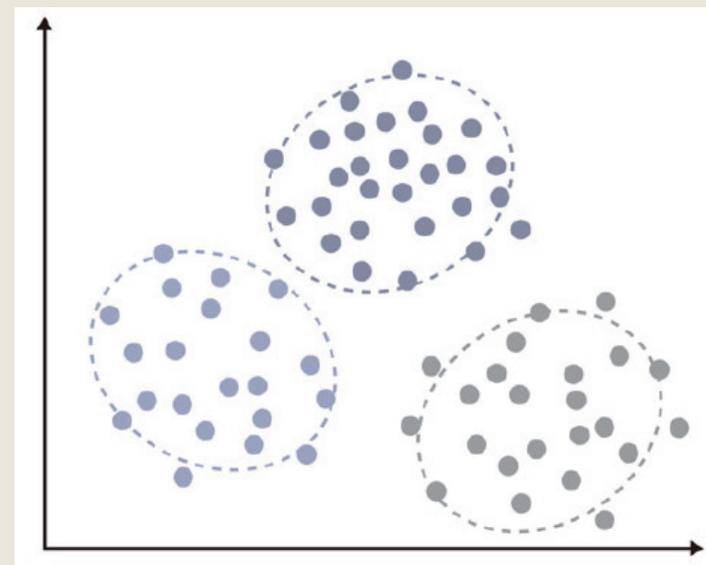


図10を見ると、なんとなく3つのグループに分けられることがわかります。

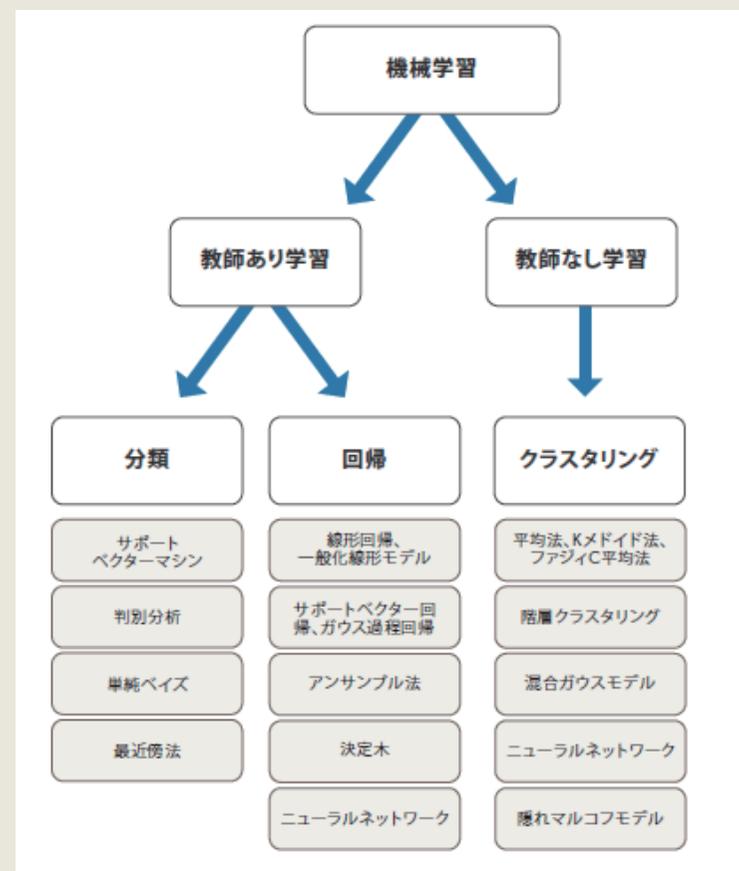
## 第2章 機械学習のタイプ

### 2-5. 教師あり学習と教師なし学習のアルゴリズム

教師あり、教師なしの機械学習アルゴリズムは何十種類もあり、学習方法もそれぞれ異なります。最も優れた手法や、何にでも使える手法というものはないのです。

適切なアルゴリズムを探すには、試行錯誤に頼らざるを得ない部分があります。極めて経験豊富なデータサイエンティストでも、ある**アルゴリズム**がうまく機能するかどうかは試してみないとわからないのです。

図11 アルゴリズムの例



## 第2章 機械学習のタイプ

### 2-6. 強化学習

強化学習に「教師」はいません。ある環境にエージェントを置き、エージェントは行動の選択を行うことができるものとします。そして、エージェントが選択した行動に対して、その環境からプラスあるいはマイナスの報酬をエージェントに与えます。

エージェントは学習の過程を通じて、**最も報酬が得られる方向に調整**していきます。その結果、問題が解決できるようになるのです。

図12 強化学習の図解



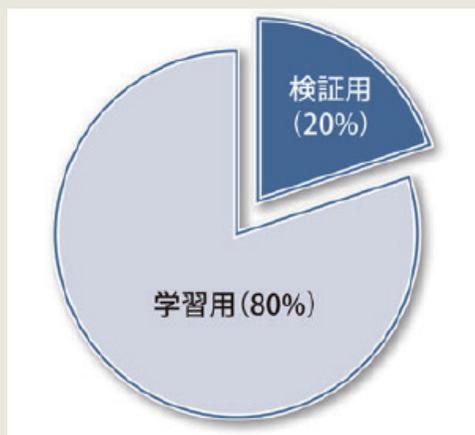
## 第2章 機械学習のタイプ

### 2-7. 機械学習のためのデータ

機械学習にとってデータはとても重要です。データサイエンティストの仕事の80%が機械学習のデータの用意に費やされていると言われていたほどです。

機械学習のために用意された学習データ（訓練データとも言います）の質が、学習の結果に大きく影響します。正確な画像認識や音声認識を実現するには、数千、数万の学習データが必要ですが、最近の優秀なアルゴリズムでは少ない学習データでも高い認識率を達成するものの開発されています。

図13 データセットにおける学習データと検証データの分け方



**Point** 学習データが少ないと結果が悪くなる

機械学習において、検証データを用意して、学習の結果を検証することはとても大切です。この割合でなければいけないというものではありませんが、学習データを8割程度、検証データを残り2割程度のデータセットを用意することが一般的です。

### 2-8. 学習データで注意すべき点

#### ■ 過学習 (over fitting)

過学習とは、学習データだけが持つ特徴に対して過剰な学習が行われることにより、実際に予測するデータに対する正解率が下がるというものです。過学習を防ぐためには、**ドロップアウト**という手法があります。

ドロップアウトとは、特定のレイヤーの出力を学習時にランダムで0に落とすことで、一部のデータが欠損していても正しく認識ができるようにするものです。これにより、画像の一部の局所特徴が過剰に評価されてしまうことを防ぎ、モデルのロバスト性（外部の影響により変化することを阻止する性質）を向上させることができます。

#### ■ 次元の呪い (the curse of dimensionality)

データの次元とは、データの特徴量のことで、データの次元数が大きくなり過ぎると、データで表現できる組み合わせも飛躍的に多くなってしまい、その結果、十分な学習結果が得られなくなります。このような現象を「次元の呪い」と言います。

関係性のない特徴量を学習させても、そもそも意味がありません。データはただ多ければいいのではなく、無駄な計算が発生しないよう、**学習する前のデータの特徴量を吟味して次元削除**することも必要です。

## 第2章 機械学習のタイプ

### 2-9. データの入手

機械学習のためにデータはどう用意するのか。ここでは、基本的な用意の方法を紹介します。

#### ■データの用意の方法

自分でデータを作る	趣味や研究であれば、時間はかかりますが、データは自作することができます。会社の業務用でしたら、いままで自社で蓄積したデータを利用することができます。
インターネットスクレイピング	インターネットスクレイピングとは、ブラウザに表示されるデータだけでなく、Webサイトのソースコードからデータを抽出し、それらを保存することを言います。 ただし、インターネットスクレイピングの合法性を取り巻く法的環境は進化し続けています。また、大量のデータをスクレイピングしようとする、相手のサーバに大きな負荷を与え、IPアドレス等でスクレイピングを行っているところを把握し、IPアドレス制限によりアクセスできないようにしているところもあります。
公開データの活用	有償・無償を問わず、コンペサイトのデータを利用するものです。日本ではSIGNATE ( <a href="https://signate.jp/">https://signate.jp/</a> ) が有名です。また、有償・無償を問わず、公開しているデータをAPIを利用して提供しているサービスもあります。

## 第2章 機械学習のタイプ

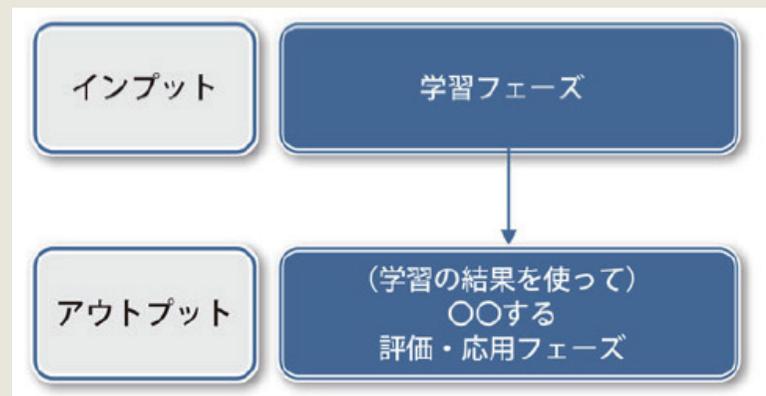
### 2-10. 教師あり学習の流れ

機械学習の流れを単純化すると、**学習フェーズ**と**評価・応用フェーズ**の2つに分けることができます。

「学習のフェーズ」では、大量のデータを学習させて、その結果として学習済のモデルができあがります。できあがった学習済モデルを「評価・応用フェーズ」で分類などの作業を行います。

例えば顔認識であれば、その顔を特定して、入力した写真やビデオからその顔を見つけます。手書き文字であれば、その手書き文字を実際に認識して、目的とする処理をしていくフェーズです。

図14 教師あり学習の流れの図解



## 第2章 機械学習のタイプ

### ■ 課題の定義

教師あり学習を行うにあたり、まずは課題の定義からスタートします。

#### ① 課題の定義

どんなことを実現したいか、何を学習させて、何を達成したいかという課題を明確にしなければいけません。研究プロジェクトであれば、プロジェクトの終了の判定を考えなければいけません。会社の企画であれば、どんな課題を解決して、何を達成したいのかを明確に定義しないと失敗する可能性が高くなります。

#### ② データの定義

課題を解決するためにどんなデータが必要になるのか、その必要となるデータに合わせて、データにつけるラベルや、その後の評価方法も一緒に検討しておくといでしょう。

## 第2章 機械学習のタイプ

### ■データの準備

機械学習の中でデータはとても重要な意味を持ちます。

次のようにデータセットを作成し、データの収集と前処理整形を行います（図15）。

#### ①自分でデータを収集・作成

独自の課題で、独自の研究開発であれば、自分で学習用のデータセットを作成する必要があります。

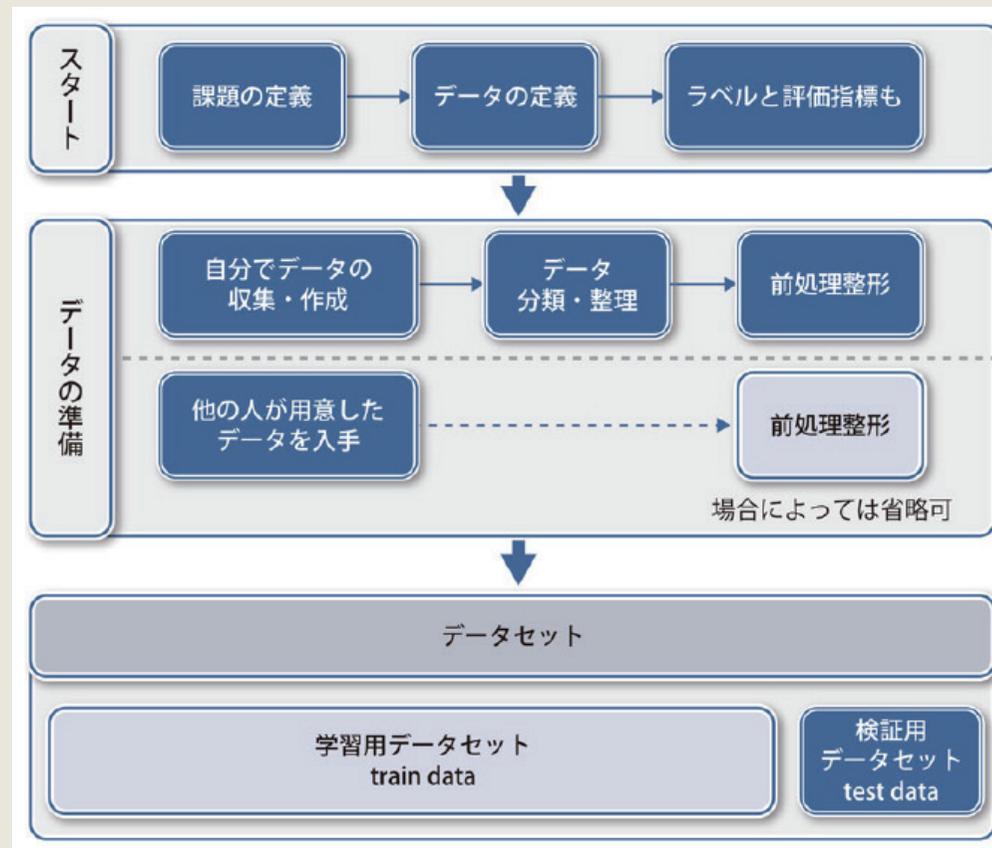
学習データを収集し、学習データの特徴を抽出しておきます。機械学習において、効果的学習を実現するための特徴量を抽出、整備することを特徴量エンジニアリングと言います。この段階が一番時間がかかります。画像認識分野では、数万枚写真のデータを用意しなければいけない場合もあり、それらの大量のデータの処理やラベリングには膨大な処理時間が必要です。

#### ②他人が用意したデータを入手

学習研究のための共通のテーマであれば（花の認識、手書き数字の認識）、他の人があらかじめ用意してくれたデータセットを活用します。データの前処理が不要な場合が多いです。

## 第2章 機械学習のタイプ

図15 教師あり学習の流れの詳細図解

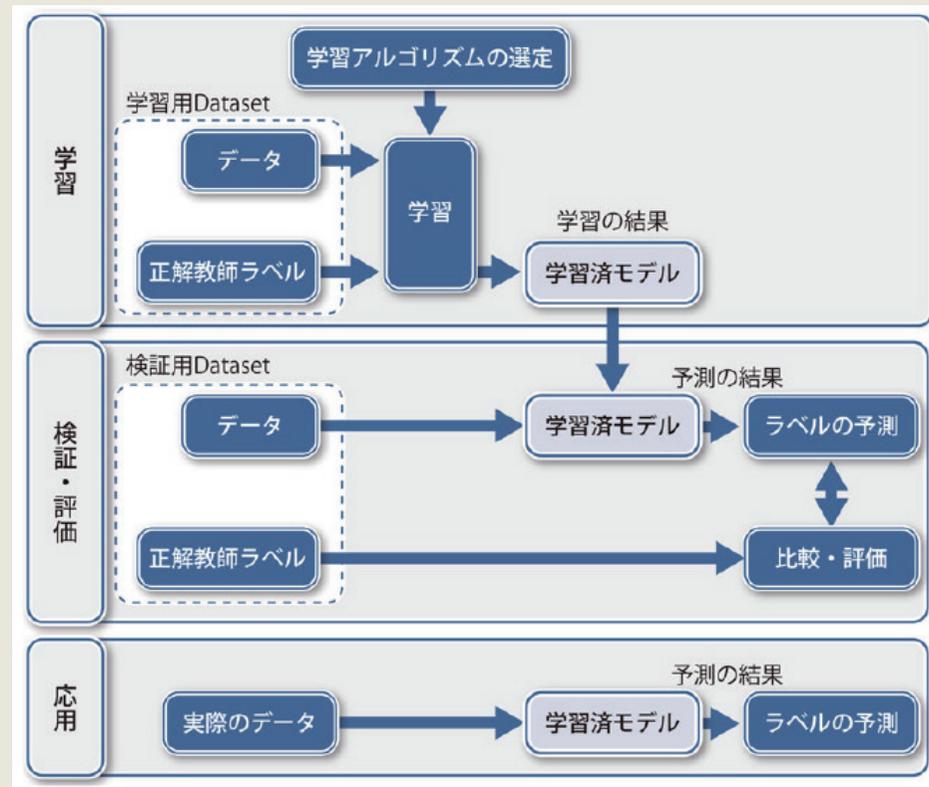


## 第2章 機械学習のタイプ

### ■学習フェーズ

学習フェーズでは、学習アルゴリズムでデータを学習させます（図16）。

図16 学習フェーズと応用フェーズ



## 第2章 機械学習のタイプ

### ■ 検証・評価フェーズ

評価では、学習フェーズで学習した結果としてのモデルを用いて、予測させて評価します。

### ■ 応用フェーズ

図17に示す機械学習における処理フローは多くのレシピにおいて適用できるフローです。この図を理解しておくことにより、レシピを試してみるときに「今、どの段階の何をやっているのか」がわかり、戸惑わずに作業がスムーズに進められるようになります。

## 第2章 機械学習のタイプ

図17 各フェーズの実施箇所



## 第3章 ディープラーニング

### 3-1. ディープラーニング（深層学習）とは

## 深層学習（DL） = Deep Learning

ディープラーニング（深層学習）とは、人間が自然に行うタスクをコンピュータに学習させる機械学習の手法のひとつです。人工知能（AI）の急速な発展を支える技術であり、その進歩によりさまざまな分野への実用化が進んでいます。近年開発の進んでいる自動運転車においてもカギとなっているのは、ディープラーニングです。

ディープラーニングの応用はさまざまですが、非常に重要なのは、**画像と映像の認識**となります。人間の外界から取り込む情報の約90%が視覚情報だと言われています。画像や映像の情報をいかに効率よく処理することが重要な意味を持ちます。

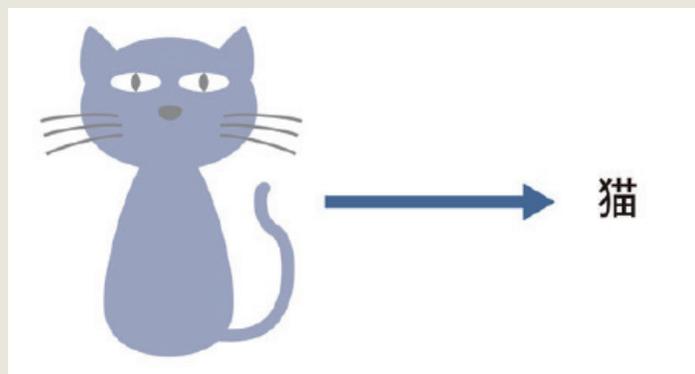
## 第3章 ディープラーニング

### 3-2. 視覚情報の重要性

人間が外界から得る情報の約90%は目から得るいわゆる「視覚情報」です。それ以外は、聴覚、触覚から来た情報で構成されていると言われています。脳がいかにかこの視覚情報を処理するか、**視覚情報**の処理をいかに**コンピュータにシミュレーション**させるかが人工知能の研究においては重要な意味を持ちます。

ディープラーニングにおける視覚情報の処理の中心になるのは、画像認識、画像分類、物体検出などになります。画像認識では特に「**画像の分類 (image classification)**」、例えばある画像や写真を見て、それは「猫」ですというイメージが経験値からの分類であることを認識しましょう。

図18 画像の分類



## 第3章 ディープラーニング

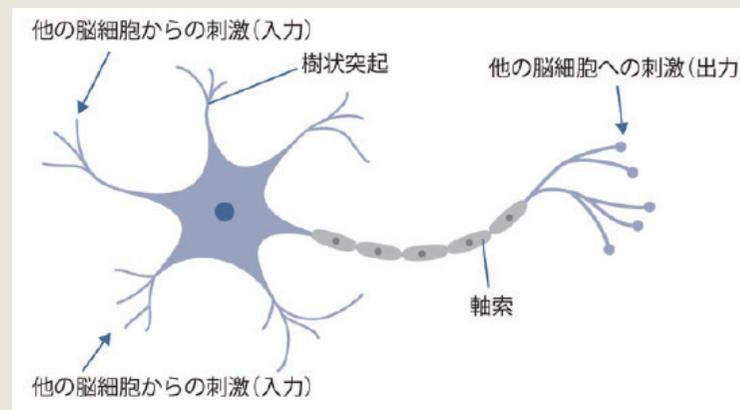
### 3-3. 生物学ニューロンから人工ニューロンへ

人工知能の研究分野で大躍進をもたらした**ディープラーニングの最小単位は人工ニューロン**です。人工ニューロンは、生物学ニューロンからヒントを得て考案されたものです。生物学ニューロンから人工ニューロンへたどり着いたのは、新しい時代の幕開けを意味します。

#### ■生物学ニューロンとは

生物学ニューロンとは、脳神経細胞のことです。簡単に示すと、樹状突起が他の神経細胞と複雑に絡み合っ  
て脳内の神経細胞のネットワークを形成しています。

図19 脳細胞の概念図



## 第3章 ディープラーニング

### ■生物学ニューロンのモデル化

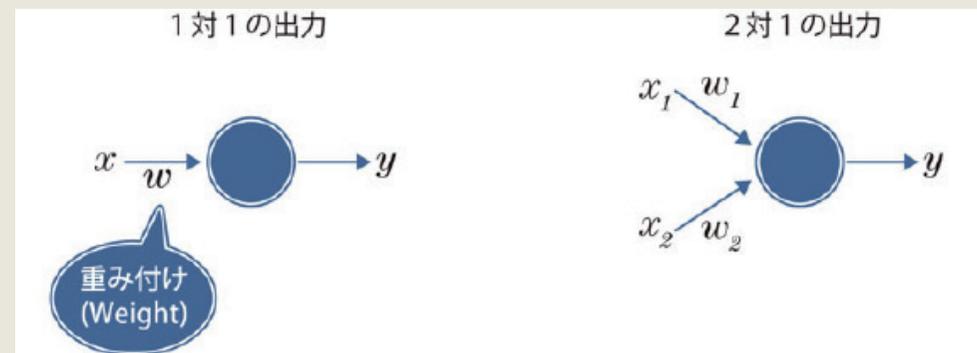
脳は他の脳細胞からの刺激を受けて、その刺激がシナプスを通して次の脳細胞へ伝播していきます。そしてモノを認識したり、状況を理解したり、さまざまな日常活動をしているのです。

この生物学ニューロンを参考にして考案されたのが、神経生理学者のウォレン・マカロック氏と数学者のウォルター・ピッツ氏のマカロック・ピッツモデルです。形式ニューロン (formal neuron) とも言います。

この形式ニューロンは非常にシンプルで、ニューロンの動きをそのまま数学的にも表現しています。

また、マカロック・ピッツモデルは演算処理 (AND、NOT、NAND、OR、XOR) を表現することもできます。

図20 ニューロンの概論図



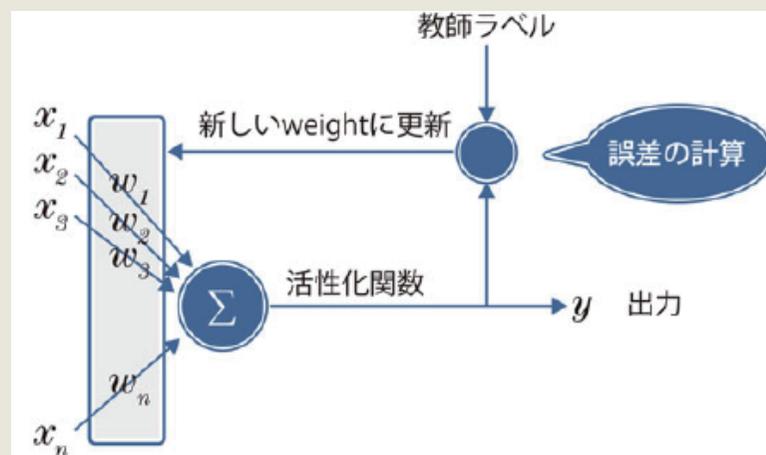
## 第3章 ディープラーニング

### ■パーセプトロン (perceptron)

パーセプトロンは1957年にアメリカの心理学研究者フランク・ローゼンブラッド氏に考案された形式ニューロンに基づいたアルゴリズムです。パーセプトロンは形式ニューロンの実装となります。人工ニューラルネットワークと、ディープラーニングの起源と基礎となっています。

パーセプトロンは、何かの入力を受けて、それに反応して、何かを出力する構造です。ただし、重み付け「w」は人間が事前に決めるのではなく、学習の結果と教師データを合わせて、その誤差をフィードバックして重み付け「w」を更新していくシステムになっています。その後、**活性化関数 (activation function)** を通して出力が決められます。

図21 パーセプトロンの学習



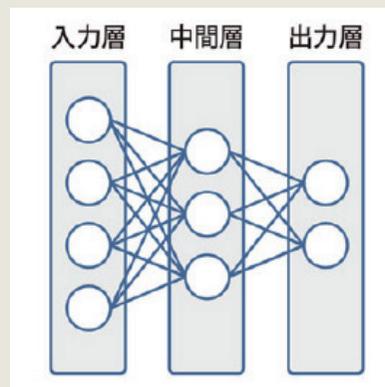
## 第3章 ディープラーニング

### 3-4. ニューラルネットワーク

パーセプトロンは機械学習において重要な革新を与えましたが、線形分離可能な課題に限られます。線形分離不可能な問題を解くためには、複数のパーセプトロンを階層的に構成する必要があります。

このパーセプトロンの限界に対応する形で、**多層パーセプトロン**の学習が提案されました。代表的な多層パーセプトロンの構成は、**入力層**、**中間層**、**出力層**の3層構造です。この構成は、ニューロンがネットワーク構造になっており、ニューラルネットワークとも言われます。

図22 多層パーセプトロン



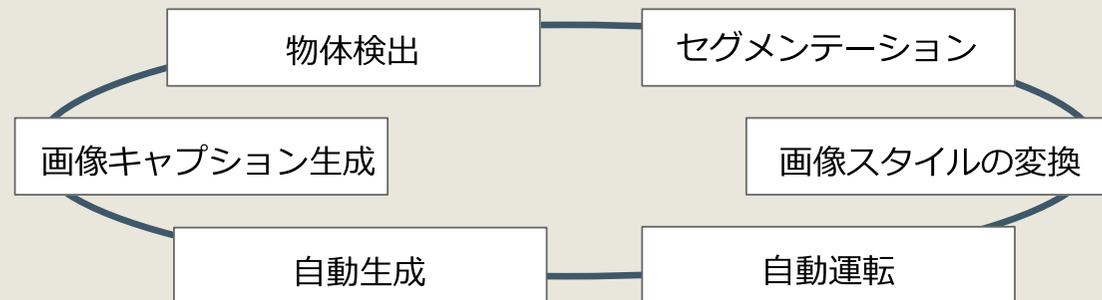
## 第3章 ディープラーニング

### 3-5. 実用化が進むディープラーニング

ディープラーニングは階層を深くしたニューラルネットワークです。一般的に**4層以上のニューラルネットワーク**を指すことが多いです。現在ではディープラーニングに対する関心が高まり、**文字認識、顔認識、自動運転、ロボットビジョン**などさまざまな分野で実用化が進んでいます。

ディープラーニングは、その文字が示すように近年の研究ではニューラルネットワークの階層がより深くなっています。複雑なアルゴリズムでは、200、300層のものもあります。

図23 ディープラーニングの応用領域



## 第3章 ディープラーニング

### 3-6. ディープラーニングに必要な数学

機械学習含め、ディープラーニングのツールやライブラリを活用するという視点で見れば、ディープラーニングの概念や考え方を理解することで対応できます。しかし、本格的な研究者や新しいアルゴリズムを開発するエンジニアやライブラリを提供するエンジニアを目指すのであれば、以下のような数学のスキルが求められます。

- 線形代数（行列の演算）
- 解析学（微積分）
- 確率・統計

## 第4章 画像処理の基礎

### 4-1. 人工知能による画像認識処理

人工知能の技術は、次のような分野で力を発揮しています。特に製造業においては画像認識が大きな力を発揮します。検査装置の画像をもとに、人工知能が不良品の判定を行うことができるからです。

- ・文字認識
- ・文字検索、自動翻訳
- ・音声認識
- ・スマートフォンによる音声会話
- ・画像認識
- ・顔認証システム
- ・自動車の自動運転
- ・深層学習
- ・将棋や碁のゲームソフト

## 第4章 画像処理の基礎

### 4-2. 画像認識とは

「5」という数字文字のいろいろなパターンを入力画像に対し、人工知能は、「5である」という答えを出すことができます。殴り書きのような手書きの「5」や、イラスト風の「5」に対しても、正確に「5」と画像認識することができます。

図24 画像認識－文字



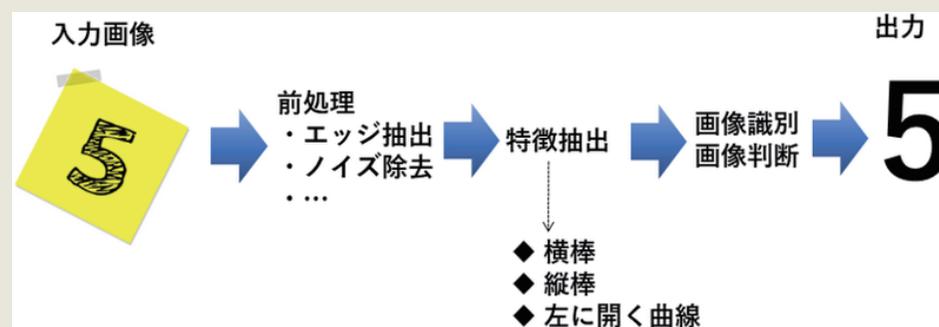
## 第4章 画像処理の基礎

### 4-3. 文字の画像認識のステップ

まず画像前処理として、画像のノイズ除去や背景除去などの①**前処理**が必要になります。続いて、入力された画像、例えば「5」には横棒と縦棒、左側に開く曲線があるという②**特徴抽出**を行います。そして、抽出された特徴から他の数字「2」「3」「7」の特徴と比較し、③**判定処理**を行います。最後に「5」と判定し、④**出力**するのです。

このようにさまざまなパターンの「5」に共通する特徴を人工知能に学習させ、正誤を判定するのが画像認識処理です。そしてこのような認識をパターン認識などと呼びます。

図25 画像認識ステップ



### 4-4. 人工知能による画像認識の方式

人工知能の学習方法には、ルールベース、機械学習、ディープラーニングといったものがあります。

- **ルールベース**

人が「5」の特徴量を教え、コンピュータが特徴量に基づき、「5」を選びます。

- **機械学習**

人がたくさんを入力画像の中から「5」の画像がどれかを教え、コンピュータはそれを元に特徴量を学習し、「5」を選びます。

- **ディープラーニング**

複数枚ある入力画像画像から、人工知能が「5」の特徴量を自ら見つけ出し、「5」を選びます。  
この際に使われる仕組みはニューラルネットワークと呼ばれます。

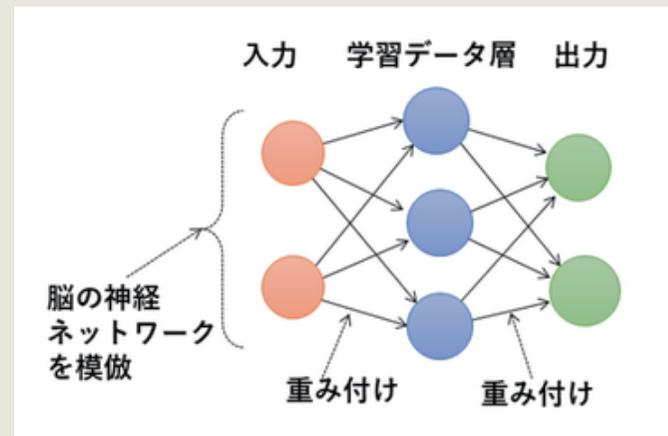
## 第4章 画像処理の基礎

### 4-5. ニューラルネットワークによる認識

ニューラルネットワークは、脳の神経ネットワークを模倣した仕組みです。

数字の「5」であれば、入力から横棒・縦棒・円弧・曲線といった情報を、重み付けを変えて各層に受け渡して処理を行い、正しい回答が出力されるように調整していきます。そのプロセスを繰り返すことにより、「5」という数字の特徴量を定義するのです。

図26 ニューラルネットワーク

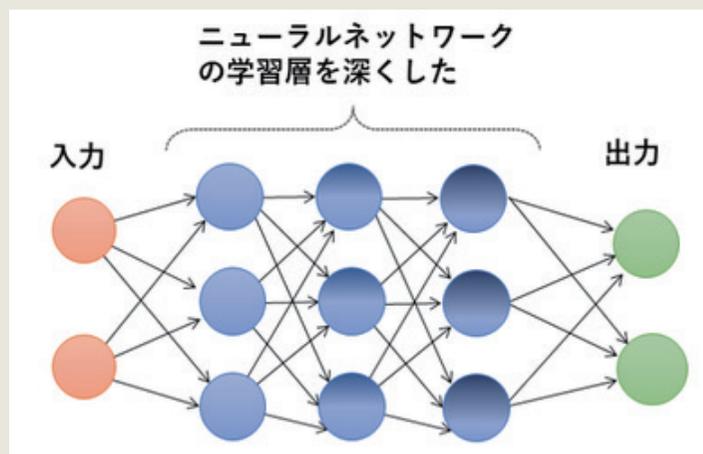


### 4-6. ディープラーニングによる認識

ディープラーニングは、学習層をさらに増やした方式で、コンピュータ性能の向上によって実現可能となりました。

より複雑な経路をたどることにより、例えば「猫と犬の見た目の違いの定義」「リンゴとナシの見た目の違いの定義」といった「言語化できない違い」を人工知能自らが学習し、見分けることができるようになったのです。

図27 ディープラーニング



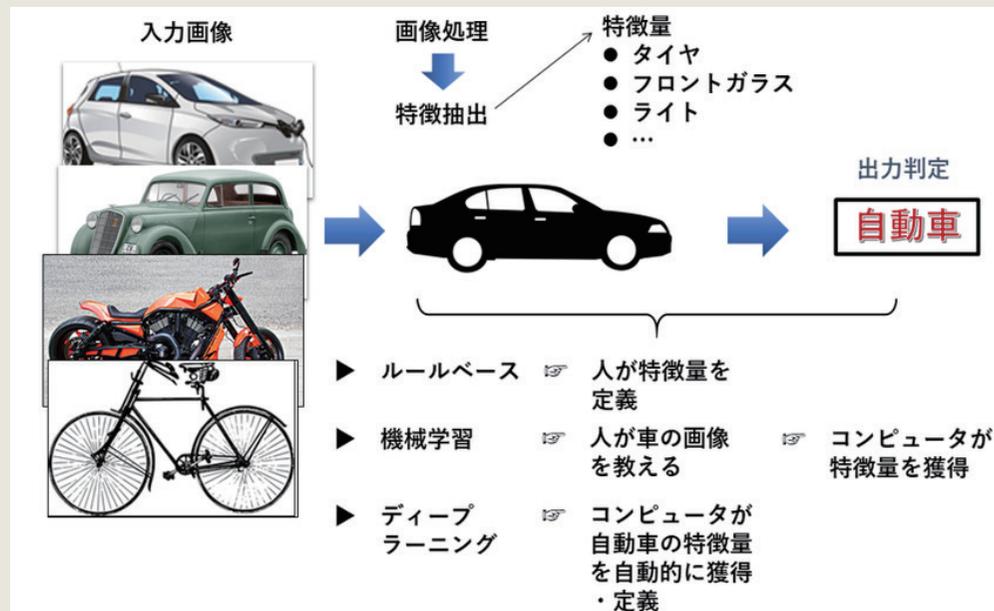
## 第4章 画像処理の基礎

### 4-7. 人工知能を使った画像処理

図28は、人工知能による画像処理について具体的に紹介したものです。

ルールベースでは大量の画像があると定義付けに対応できない問題が、大量の画像を扱う必要がある、人による対応の手間が増加するという問題があります。

図28 ディープラーニング



## 第5章 Pythonプログラミング

### 5-1. Pythonとは

Pythonとは1991年にオランダ人のガイド・ヴァンロッサムというプログラマーによって開発され、オープンソースで運営されているプログラミング言語です。その用途は、組み込み開発、Webアプリケーション、デスクトップアプリケーション、人工知能開発、ビッグデータ解析などと多岐に渡ります。

Pythonのプログラミング言語としての主な特徴は、少ないコードで簡潔にプログラムを書けること、専門的なライブラリが豊富にあることなどが挙げられます。

Pythonのプログラミングが簡単な理由の一つに「オフサイドルール」初稿というコードの書き方の規則があります。「オフサイドルール」初稿とは行頭をインデント（字下げ）することによって文の塊を指定するというプログラミング言語の規則です。開発の現場において、他の人が書いたコードを読めるということはとても重要なことなのです。

また、Pythonは開発に役立つプログラムをまとめたものである「ライブラリ」の数が数万に上ります。こうした既存のライブラリを有効に活用することで、自分の作りたいプログラムを作成することが容易にできるのも特徴です。

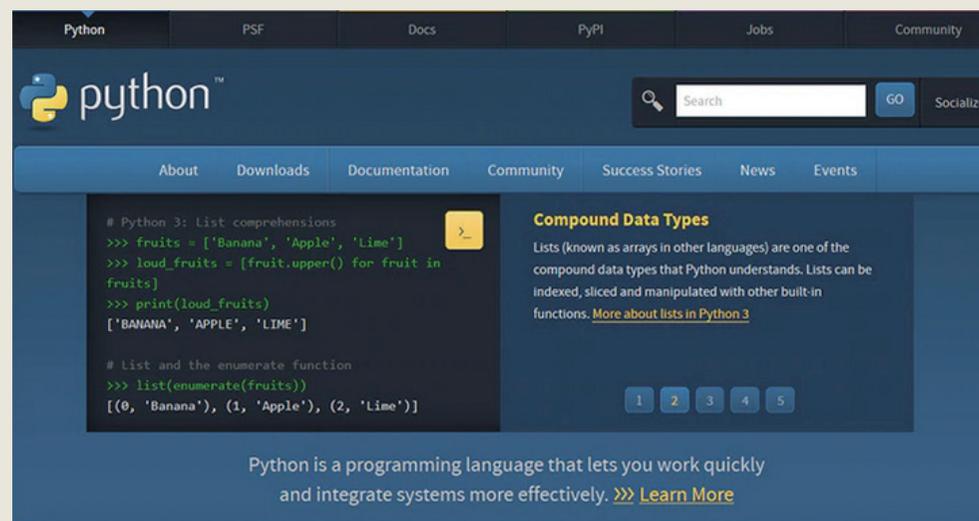
## 第5章 Pythonプログラミング

### 5-2. 数多くのWebアプリケーションの開発に活用

Pythonは数多くのWebアプリケーションの開発に活用されています。

代表的なWebアプリケーションとしては、Instagram、YouTube、Evernote、DropBoxなどです。いずれも日本だけでなく世界中で使用されているアプリケーションです。

このように、最新のWebアプリケーション開発にPythonが選ばれている理由には、読み書きがしやすいこと、コードの記述量が少なく済むこと、C言語との相性がよいことなどが挙げられます。



Pythonのサイト (<https://www.python.org/>) から最新バージョンがダウンロードできます。

### 5-3. 人工知能の開発に活用

Pythonは、人工知能を搭載したソフトウェア、とりわけ機械学習を用いたソフトウェアの開発の分野で広く活用されています。

コードが簡潔であるだけでなく、初めて機械学習を学ぶ人にとっても習得しやすい言語であるため、機械学習を学ぶ際には基本のプログラミング言語といえます。さらに、Pythonは統計処理や数値計算を得意とするプログラミング言語です。こうした特性もPythonが機械学習やディープラーニング、さらにはデータ解析の処理によく用いられる理由のひとつです。

具体的にはTensorFlow（テンソルフロー）という機械学習ライブラリがディープラーニング・ニューラルネットワークの分野で活用されています。

#### Point TensorFlow が活用される理由



TensorFlowは、機械学習向けに開発されたエンドツーエンドのオープンソースプラットフォームです。研究者が機械学習で最新の実験を行い、デベロッパーがML搭載アプリケーションを簡単に開発してデプロイできるよう、各種ツールやライブラリ、コミュニティリソースを備えた総合的で柔軟性に富んだエコシステムが確立されています。

## 第5章 Pythonプログラミング

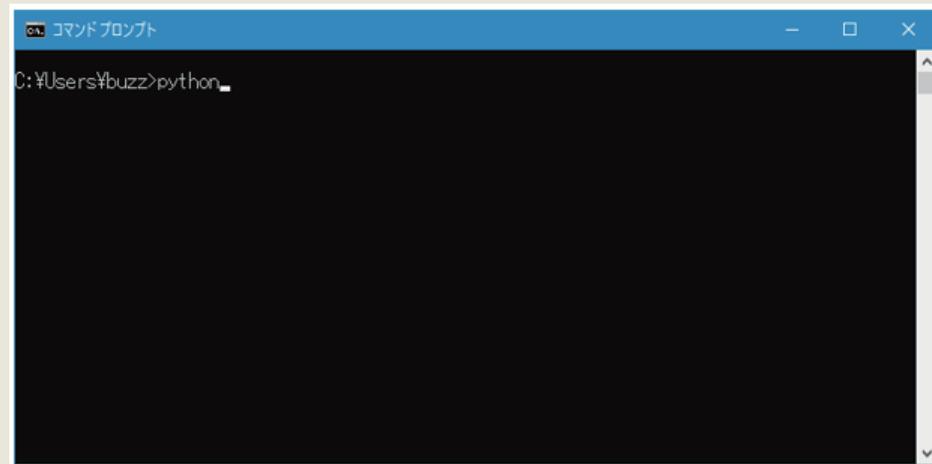
### 5-4. Pythonプログラムの基本事項

Pythonを使って記述されたプログラムを実行する方法や、プログラムの記述方法などPythonを使ったプログラミングを行う上で基本となる項目に関して解説します。

#### 1. Pythonを対話モードで利用

Pythonでは入力したコードをすぐに実行することができる対話モードで利用する機能が用意されています。ここではPythonを対話モードで起動し利用する方法について解説します。

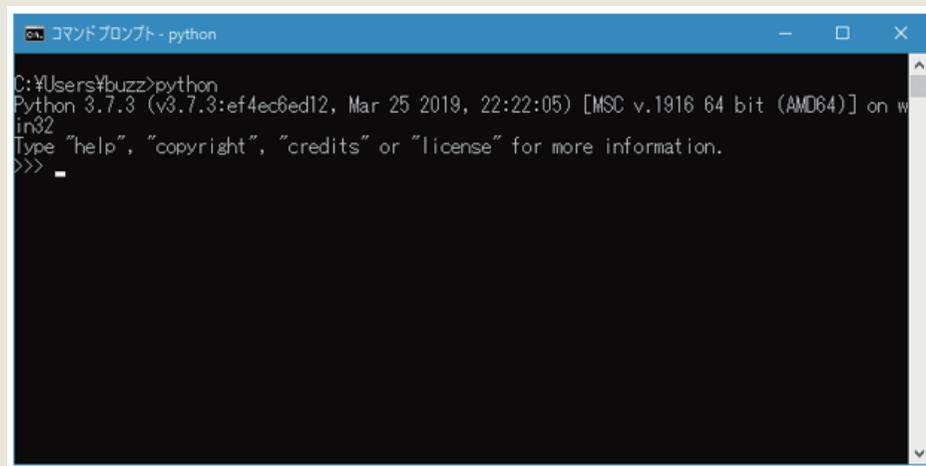
- ① Windows環境でPythonを対話モード（インタラクティブモードともいいます）で起動するには、コマンドプロンプトを起動したあとでpythonと実行します。



```
コマンドプロンプト
C:\Users\buzz>python_
```

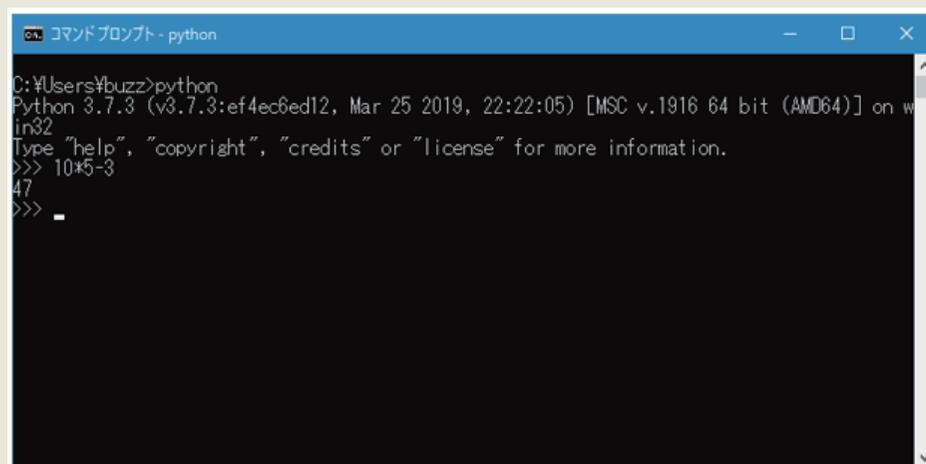
## 第5章 Pythonプログラミング

② 入力待ちの画面になります。



```
コマンドプロンプト - python
C:\Users\buzz>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> _
```

③ 最初に簡単な四則演算を行ってみます。「10\*5-3」を入力して[Enter]キーを押します。演算した結果である「47」が画面に表示されます。



```
コマンドプロンプト - python
C:\Users\buzz>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 10*5-3
47
>>> _
```

## 第5章 Pythonプログラミング

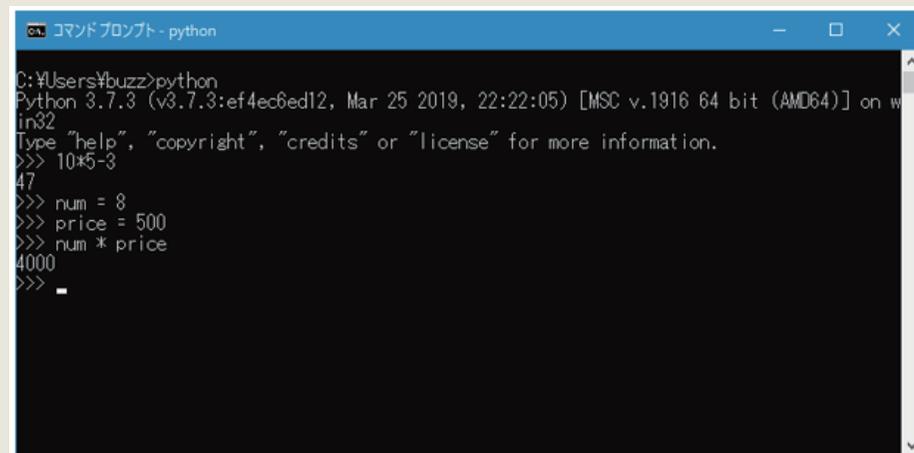
- ④ 次に簡単な変数を使った処理を行ってみます。

```
num = 8
```

```
price = 500
```

```
num * price
```

その結果である「4000」が画面に出力されます。

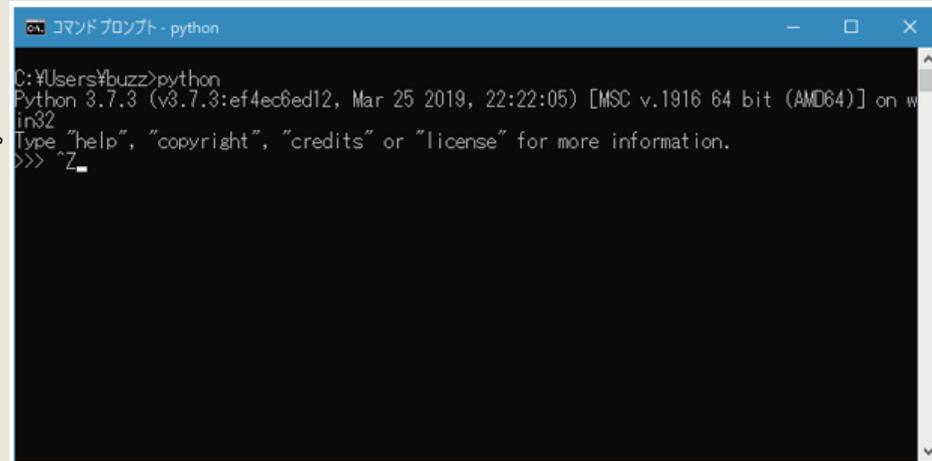


```
コマンドプロンプト - python
C:\Users\buzz>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 10*5-3
47
>>> num = 8
>>> price = 500
>>> num * price
4000
>>> -
```

Pythonでは、単なる計算といったものだけでなく、変数や条件分岐、繰り返し処理などを使ったものまで、スクリプトをファイルなどに記述することなく実行し、結果を得ることができます。

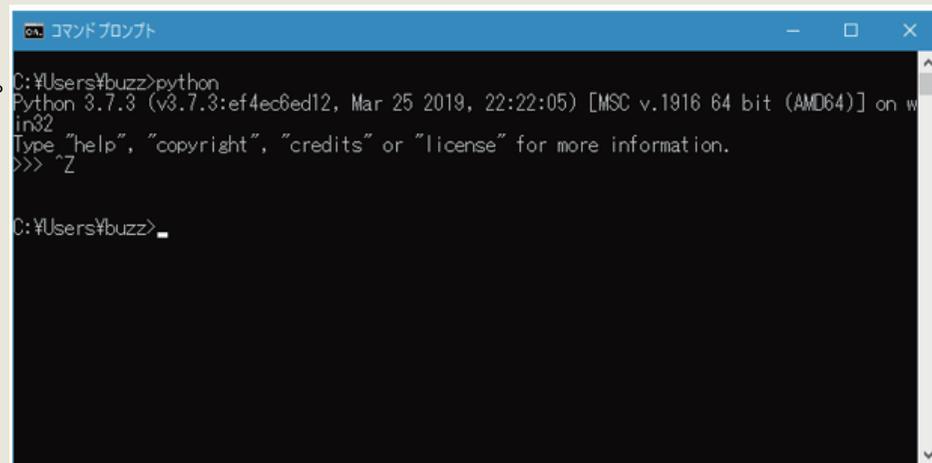
## 第5章 Pythonプログラミング

- ⑤ 対話モードを終了するには、  
Windows環境の場合は「Ctrl」  
+ 「z」キーを押してください。



```
コマンドプロンプト - python
C:\Users\%buzz>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ^Z
```

- ⑥ 「Enter」キーを押してください。対話モードが終了します。



```
コマンドプロンプト
C:\Users\%buzz>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ^Z
C:\Users\%buzz>
```

## 第5章 Pythonプログラミング

### 2. ファイルに記述したプログラムを実行

Pythonではあらかじめテキストファイルに保存しておいたプログラムを読み込んで実行させることができます。プログラムをファイルに保存しておけば、繰り返し同じプログラムを実行することができ、また修正も簡単に行うことができます。

Pythonを使ってプログラムを記述する場合は、テキストファイルに記述します。エディタはWindows標準搭載のメモ帳でも保存できますが、文字コードとしてUTF-8が使用できるエディタを使用してください。

- ① 今回は例として次のように入力します。

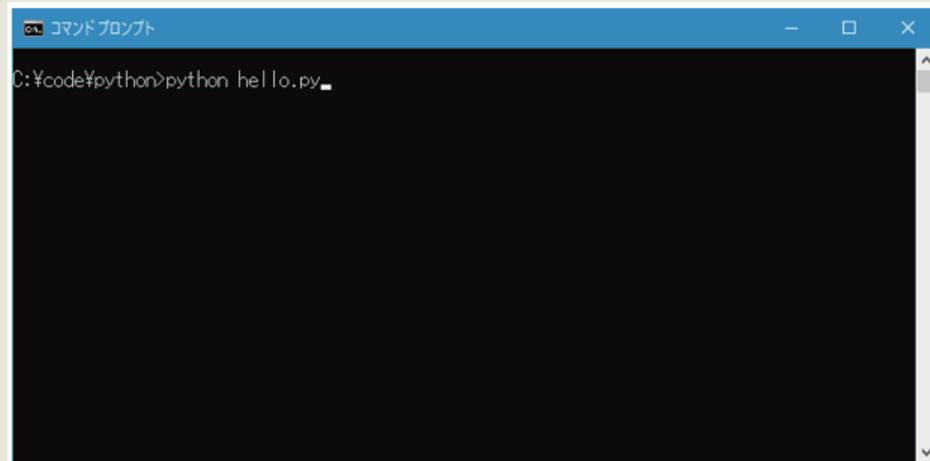
```
print("Hello, Python")
```

保存方法はメモ帳などエディタの保存方法をそのまま利用できます。ただし、拡張子は「.py」にします。



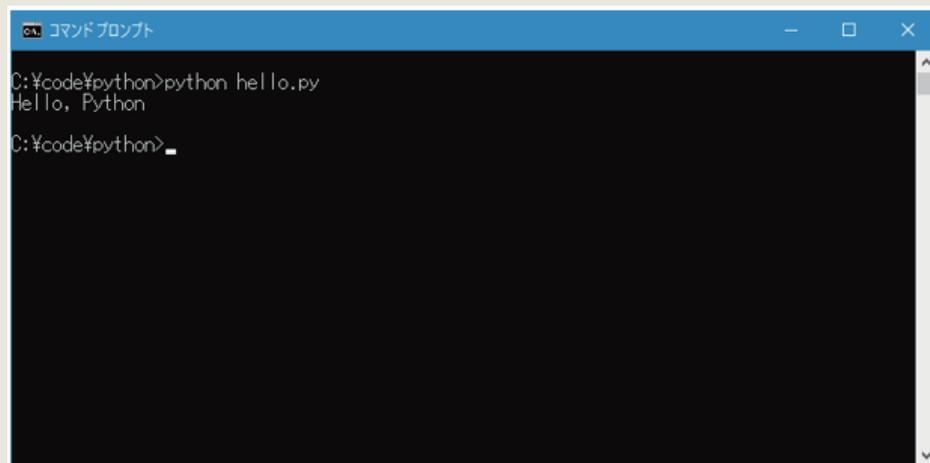
## 第5章 Pythonプログラミング

- ② ファイルに保存したプログラムを実行するには、コマンドプロンプトを起動し実行するファイルが保存されているディレクトリへ移動してください。それから「ファイル名.py」を指定します。



```
コマンドプロンプト
C:\>python hello.py
```

- ③ ファイルに記述されたスクリプトが順に実行されます。先ほど保存したプログラムが実行されて画面に  
Hello, Python  
と表示されました。



```
コマンドプロンプト
C:\>python hello.py
Hello, Python
C:\>
```

## 第6章 データの前処理技術

### 6-1. 人工知能の実装プロセス

人工知能の実装には、大きく「要件定義」「データ前処理」「学習モデルの作成・評価」「実装」の4つのステップがあります。

最初の要件定義では、主に自社のビジネス課題の把握から人工知能で実装する内容を決定し、人工知能プロジェクトのKPIを設定します。

次にデータの前処理です。データの前処理では、人工知能で利用するデータを準備し、データの品質を高めるためにクレンジングを行います。このようにデータの品質を高めた上で、そのデータを基に分析設計を行い、学習モデルの作成・評価に入ります。

学習モデルの作成・評価では、予測の精度が低い場合、何度も何度も学習を重ねて精度を高める必要があります。そうして何度も学習した精度の高いモデルを実装します。

実装段階では、構築したモデルの品質を保つために、再学習を行う必要があります。

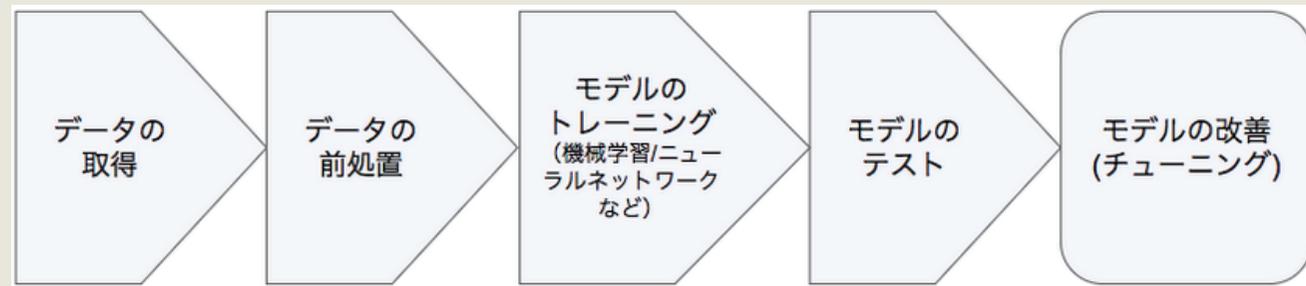
この流れの中で一見シンプルに見えるデータの前処理ですが、実は実装までの時間のうち、**7~8割がデータの前処理**に割かれるとされています。

### 6-2. 機械学習に向けてのデータ前処理

機械学習では、まずデータを用意し、それを適切なアルゴリズムに与えて学習させる、というプロセスを経るのが一般的です。その際に、用意したデータをそのままアルゴリズムに与えてうまく学習できることはまれです。実際にはアルゴリズムに入れる前に、データの前処理というものが必要になります。

データの前処理とは、得られたデータを何らかの機械学習アルゴリズムに入れる前に、そのデータに何かしら手を加えることを言います。用意した「生」のデータをアルゴリズムが学習できる、あるいは学習しやすいようにきれいなデータに整える、といったようなイメージです。

図29 機械学習データ処理のプロセス



### 6-3. データの前処理はなぜ必要か

人工知能実装にはコンピュータに読み込ませるデータがまず必要になりますが、用意したデータが完璧である可能性はほぼありません。ほとんどのデータには欠損、ノイズ、エラー値などがあります。

さらに生のデータというのは、今の値の見え方がそのデータの内部構造や関係性を表すのにベストであることは滅多にありません。最大限にそのデータを活用するには、何らかの統計的処理を施す必要があります。また、機械学習アルゴリズム自体にも、利用するデータのフォーマットに相性があります。ニューラルネットワークに画像を流すときも、通常は流せる画像のサイズは固定なので、大きすぎたり小さすぎたりする画像はリサイズする必要があります。

以上のような理由から、事前に得られた生のデータをできるだけ構造や関係性がつかみやすく、かつ機械学習アルゴリズムが学習しやすいような形に整えてあげる必要があるのです。

図30 データに潜む課題



- ・データにノイズや異常値、欠損がある
- ・不要なデータが含まれている
- ・データの形式がバラバラになっている
- ・そのまま利用できないデータがあり

### 6-4. データの前処理

「**データクレンジング**」とはその名の通り、データをきれいにする（クレンジングする）という段階になります。データのエラーやノイズ、欠損値などに対応するのがこの作業です。一見簡単そうに見える作業ですが、データの一部ごとに取り除いてしまうか、中央値を取るかなど簡単なものから、もっと複雑なものもあり、操作によって後の機械学習に影響を与えてしまうため、ある程度の機械学習における知識・経験が必要になってきます。

データクレンジングのあとに来る作業が「**データの統合**」です。データがさまざまなデータソースから収集され、データが結合されて一貫したデータが形成されます。

モデルの必要性に応じて生データを指定のフォーマットに変換するプロセスが「**データの変換**」です。データの変換には一例として以下のような方法があります。

- ・ **データの正規化**

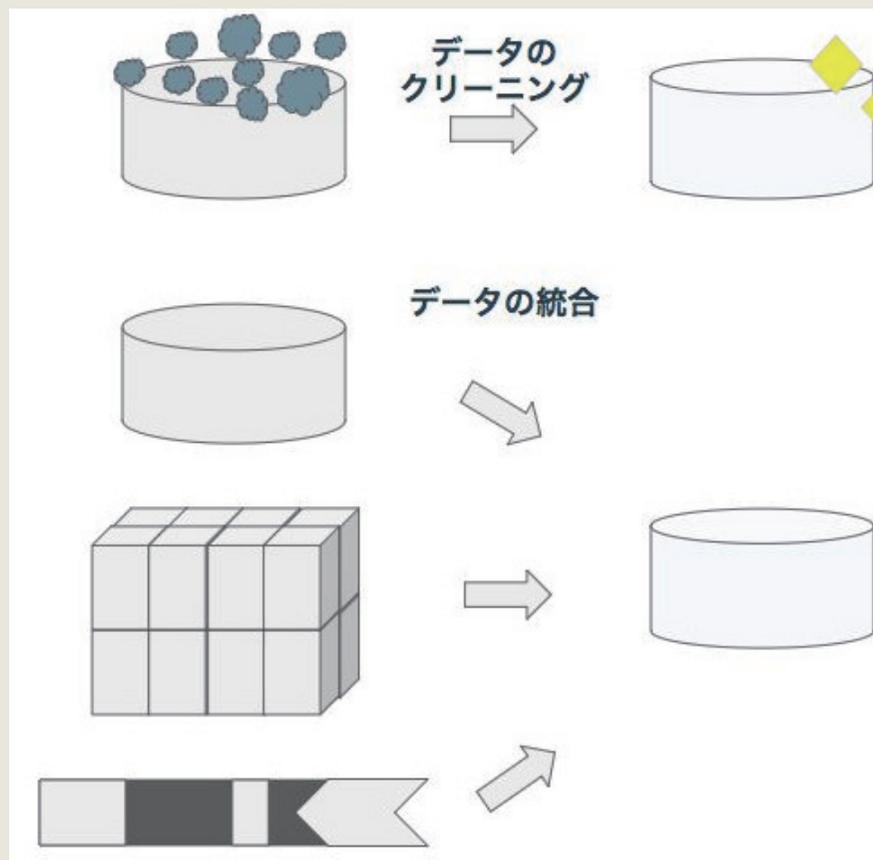
主にデータのスケールリングを実行できるように、数値データを決められた範囲、例えば0から1の範囲に変換することです。

- ・ **汎化**

特定の細かい分野ではなく、一般化することです。

## 第6章 データの前処理技術

図31 データ前処理のイメージ



### 6-5. 機械学習のためのデータ前処理

機械学習は、データ内の複雑でかつ有用な可能性があるパターンを自動的に見つけ出すのに役立ちます。こうしたパターンは機械学習モデルに凝縮し、それを新しいデータポイントで使用できます。これは「予測の作成」または「推論の実行」と呼ばれるプロセスです。

機械学習モデルの構築は複数の段階からなるプロセスです。各段階にはそれぞれ独自の技術的および概念的な課題があります。この2部構成の記事では、ソースデータを選択、変換、拡張して、ターゲット（応答）変数への強力な予測信号を作成するプロセス（教師あり学習タスク）に焦点を当てます。こうしたオペレーションは、ドメインに関する知識とデータサイエンスの技術を組み合わせて行われます。これが特徴エンジニアリングの中心となるものです。

機械学習のためのデータ前処理では、**データエンジニアリング**と**特徴エンジニアリング**の両方が必要になります。データエンジニアリングは元データを準備済みデータに変換するプロセスです。次に特徴エンジニアリングでは、準備済みのデータを調整し、機械学習モデルによって求められる特徴を作成します。

## 第6章 データの前処理技術

### ●元データ（または単にデータ）

元データとは、機械学習のための事前準備がなされていないソース形式のデータを表します。データは未加工の形式（データレイク内）または変換された形式（データウェアハウス内）の場合があることに留意してください。

データウェアハウス内の変換済データは、分析に使用するため元の未加工の形式から変更されている可能性があります。そのデータが特に機械学習タスク用に準備されていないことを意味します。

また、ストリーミングシステムから送信されたデータで、結果的に予測のため機械学習モデルを呼び出すものは、未加工の形式のデータとみなされます。

### ●準備済データ

準備済データとは、機械学習タスクの準備が整った形式のデータセットを表します。データソースが解析され、結合され、表形式になったものです。データは集計され、適切な粒度に要約されています。

例えばデータセットの各行が一意的顧客を表し、各列はその顧客の要約情報（過去6週間に費やされた合計額など）を表す場合があります。教師あり学習タスクの場合は、ターゲットとなる特徴が存在します。無関係な列は削除され、無効なレコードは除外されています。

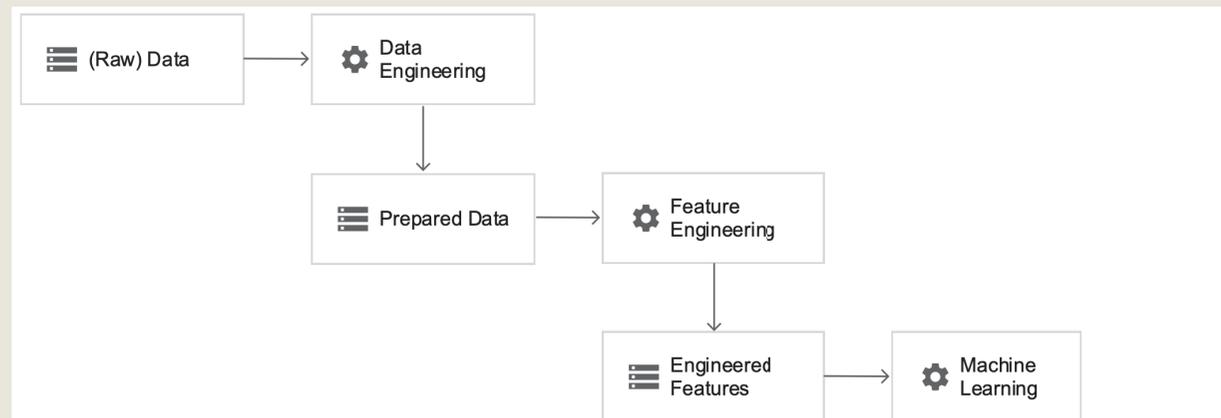
## 第6章 データの前処理技術

### ●設計済データ

設計済特徴とは、モデルによって求められる調整済の特徴を備えたデータセットを表します。つまり、準備済データセットの列で特定の機械学習固有オペレーションを実行し、トレーニング時および予測時にはモデル用の新しい特徴を作成するものです。

例としては、数値列を0～1の値にスケールリングする、値をクリッピングする、カテゴリ別の特徴をワンホットエンコードするといったことがあります。

図32 元データから準備済データ、設計済特徴、機械学習までのデータの流れ



## 第6章 データの前処理技術

### 6-6. 前処理オペレーション

構造化データの場合、データ前処理オペレーションには次のようなものがあります。

<b>データクレンジング</b>	破損した値や無効な値を含むレコードを元データから削除または訂正すること、および多数の列が欠落しているレコードを削除すること。
<b>インスタンスの選択とパーティショニング</b>	入力データセットからデータポイントを選択して、トレーニング、評価（検証）、およびテスト用のセットを作成すること。このプロセスには、反復可能なランダムサンプリング、少数派クラスのオーバーサンプリング、および層別パーティショニングの手法が含まれます。
<b>特徴の調整</b>	機械学習用の特徴の品質を高めること。これには数値の増減と正規化、欠損値の転嫁外れ値のクリッピング、傾斜分布による値の調整などが含まれます。
<b>表現変換</b>	数値特徴を（バケット化によって）カテゴリ的特徴に変換し、カテゴリ的特徴を（ワンホットエンコード、カウントによる学習、スパース特徴埋め込みなどによって）数値表現に変換すること。一部のモデルは数値特徴またはカテゴリ的特徴しか利用できませんが、その他のモデルは混合タイプの特徴を処理できます。両方のタイプを処理するモデルでも、同じ特徴の異なる表現（数値およびカテゴリ）を活用すると有効です。

## 第6章 データの前処理技術

特徴抽出	PCA、抽出の埋め込み、ハッシュ化などの技法を利用して、より低次元かつ強力なデータ表現を作成し、特徴の数を減らすこと。
特徴選択	モデルをトレーニングするための入力特徴のサブセットを選択し、フィルタまたはラッパーメソッドを使用して、無関係または冗長な特徴を無視すること。特徴に多数の値が欠損している場合は、その特徴を単純に削除する場合があります。
特徴構築	多項式展開（一変数関数を使用）などの一般的な手法か、または特徴の関連付けを（特徴量の相互作用を取得するため）使用して、新しい特徴を作成すること。特徴は機械学習ユースケースのドメインのビジネスロジックを使用して構築することもできます。

非構造化データ（画像、音声、テキスト文書など）を扱う場合、ドメインのナレッジベースの特徴エンジニアリングは、ディープラーニングによりモデルのアーキテクチャに折り込むことで排除されました。畳み込み層は自動の特徴プリプロセッサです。もちろん適切なモデルアーキテクチャを構築するには、データに関する多少の経験的知識が必要です。また、次のような若干の前処理も必要になります。

テキスト文書の場合は、ステミングと見出し語化、TF-IDF計算、およびnグラム抽出と埋め込みルックアップ。画像の場合は、クリッピング、サイズ変更、トリミング、ガウスぼかし、およびカナリア色フィルタ。特徴エンジニアリングの手順として、完全にトレーニングされたモデルの最後の層を除くすべての層を扱う転送学習。これはテキストや画像を含むあらゆる種類のデータに適用されます。

令和2年度「専修学校による地域産業中核的人材養成事業」  
Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

■実施委員会

◎ 船山 世界	日本電子専門学校 校長
大川 晃一	日本電子専門学校 エンジニア教育部長 ／ケータイ・アプリケーション科科长
種田 裕一	東北電子専門学校 第2教務部長 学生サポート室長
勝田 雅人	トライデントコンピュータ専門学校 校長
安田 圭織	学校法人上田学園 上田安子服飾専門学校
平田 眞一	学校法人第一平田学園 理事長
平井 利明	静岡福祉大学 特任教授
木田 徳彦	株式会社インフォテックサーブ 代表取締役
渡辺 登	合同会社ワタナベ技研 代表社員
岡山 保美	株式会社ユニバーサル・サポート・システムズ 取締役
富田 慎一郎	株式会社ウチダ人材開発センタ 代表取締役社長

■人材育成委員会

◎ 大川 晃一	日本電子専門学校 エンジニア教育部長 ／ケータイ・アプリケーション科科长
福田 竜郎	日本電子専門学校 AI システム科
阿保 隆徳	東北電子専門学校 学科主任
小澤 慎太郎	中央情報大学校 高度情報システム学科
神谷 裕之	名古屋工学院専門学校 メディア学部 情報学科
北原 聡	麻生情報ビジネス専門学校 校長代行
原田 賢一	有限会社ワイズマン 代表取締役
柴原 健次	合同会社ヘルシーブレイン 代表 CEO
菊嶋 正和	株式会社サンライズ・クリエイティブ 代表取締役

■評価委員会

平井 利明	静岡福祉大学 特任教授
富田 慎一郎	株式会社ウチダ人材開発センタ 代表取締役社長
平田 眞一	学校法人第一平田学園 理事長

令和2年度「専修学校による地域産業中核的人材養成事業」  
Society5.0 実現のための IT 技術者養成モデルカリキュラム開発と実証事業

人工知能教育研修プログラム教材

令和3年2月

学校法人電子学園（日本電子専門学校）  
〒169-8522 東京都新宿区百人町1-25-4  
TEL 03-3369-9333 FAX 03-3363-7685

●本書の内容を無断で転記、掲載することは禁じます。