

G検定（JDLA ジェネラリスト検定）対策

# 完全攻略テキスト

公式シラバスの全495語を、1冊で。

---

10の大項目・55の中項目・495語をシラバスの並びのまま収録

図解65点／大項目ごとの確認問題とチェックリスト

巻末に495語の総索引

## AI資格ラボ

YouTubeチャンネル登録者への配布テキストです。

出題範囲は一般社団法人 日本ディープラーニング協会（JDLA）の公式シラバスに準拠しています。

掲載している問題・解説はすべて自作であり、試験の過去問ではありません。

## はじめに この本の使い方

この本は、G検定の公式シラバスに載っている**詳細キーワード495語**を、**シラバスの並びのまま**ひととおり収めたものです。覚えやすい順に並べ替えてはいません。試験範囲のどこにいるかが分かるほうが、あとで探しやすいためです。

### この本の作り

- **10の大項目・55の中項目・495語。**中項目のはじめに「この節の目標」を置き、そのあと語を順に説明します。
- **目標のほうにも出題点があります。**シラバスの目標には、キーワードとして挙がっていない観点が含まれています。読み飛ばさないでください。
- **図解65点。**絵が説明の仕事をする所にだけ置いてあります。A4・実物大（100%）で刷ると、図の中の文字まで読めます。
- **大項目の終わりに確認問題とチェックリスト。**読んだだけで分かった気になるのを防ぐためのものです。

- ⇒ この本は、YouTubeチャンネル「AI資格ラボ」の講義動画と同じ材料から作っています。語の説明文は動画で読み上げているものと同じで、図も同じです。**動画で流れをつかみ、この本で確かめる。あるいはこの本を先に読んでから動画を見る。**どちらでも構いません。
- ⇒ **練習問題は無料の問題集サイトにあります。**大項目ごとに解けるほか、本番と同じ145問100分の模擬試験も無料・登録不要で受けられます。この本の語は、そのすべてに問題が用意してあります。

## 目次

## この本の中身

|            |                          |      |    |
|------------|--------------------------|------|----|
| <b>第1部</b> | <b>人工知能とは</b>            | 技術分野 | 5  |
|            | 1 人工知能の定義                |      | 6  |
|            | 2 人工知能分野で議論される問題         |      | 7  |
| <b>第2部</b> | <b>人工知能をめぐる動向</b>        | 技術分野 | 12 |
|            | 3 探索・推論                  |      | 13 |
|            | 4 知識表現とエキスパートシステム        |      | 14 |
|            | 5 機械学習                   |      | 17 |
|            | 6 ディープラーニング              |      | 18 |
| <b>第3部</b> | <b>機械学習の概要</b>           | 技術分野 | 22 |
|            | 7 教師あり学習                 |      | 23 |
|            | 8 教師なし学習                 |      | 26 |
|            | 9 強化学習                   |      | 28 |
|            | 10 モデルの選択・評価             |      | 30 |
| <b>第4部</b> | <b>ディープラーニングの概要</b>      | 技術分野 | 37 |
|            | 11 ニューラルネットワークとディープラーニング |      | 38 |
|            | 12 活性化関数                 |      | 39 |
|            | 13 誤差関数                  |      | 40 |
|            | 14 正則化                   |      | 41 |
|            | 15 誤差逆伝播法                |      | 42 |
|            | 16 最適化手法                 |      | 43 |
| <b>第5部</b> | <b>ディープラーニングの要素技術</b>    | 技術分野 | 50 |
|            | 17 全結合層                  |      | 51 |
|            | 18 畳み込み層                 |      | 51 |
|            | 19 正規化層                  |      | 53 |
|            | 20 プーリング層                |      | 54 |
|            | 21 スキップ結合                |      | 55 |

|            |                            |            |
|------------|----------------------------|------------|
| 22         | 回帰結合層                      | 55         |
| 23         | Attention                  | 58         |
| 24         | オートエンコーダ                   | 60         |
| 25         | データ拡張                      | 61         |
| <b>第6部</b> | <b>ディープラーニングの応用例</b> 技術分野  | <b>66</b>  |
| 26         | 画像認識                       | 67         |
| 27         | 自然言語処理                     | 72         |
| 28         | 音声処理                       | 75         |
| 29         | 深層強化学習                     | 78         |
| 30         | データ生成                      | 81         |
| 31         | 転移学習・ファインチューニング            | 83         |
| 32         | マルチモーダル                    | 84         |
| 33         | モデルの解釈性                    | 86         |
| 34         | モデルの軽量化                    | 87         |
| <b>第7部</b> | <b>AIの社会実装に向けて</b> 技術分野    | <b>91</b>  |
| 35         | AIプロジェクトの進め方               | 92         |
| 36         | データの収集・加工・分析・学習            | 95         |
| <b>第8部</b> | <b>AIに必要な数理・統計知識</b> 技術分野  | <b>97</b>  |
| 37         | AIに必要な数理・統計知識              | 98         |
| <b>第9部</b> | <b>AIに関する法律と契約</b> 法律・倫理分野 | <b>105</b> |
| 1          | 個人情報保護法                    | 106        |
| 2          | 著作権法                       | 108        |
| 3          | 特許法                        | 109        |
| 4          | 不正競争防止法                    | 110        |
| 5          | 独占禁止法                      | 111        |
| 6          | AI開発委託契約                   | 111        |

|             |                             |            |
|-------------|-----------------------------|------------|
| 7           | AIサービス提供契約                  | 112        |
| <b>第10部</b> | <b>AI倫理・AIガバナンス</b> 法律・倫理分野 | <b>116</b> |
| 8           | 国内外のガイドライン                  | 117        |
| 9           | プライバシー                      | 118        |
| 10          | 公平性                         | 118        |
| 11          | 安全性とセキュリティ                  | 119        |
| 12          | 悪用                          | 121        |
| 13          | 透明性                         | 121        |
| 14          | 民主主義                        | 122        |
| 15          | 環境保護                        | 122        |
| 16          | 労働政策                        | 123        |
| 17          | その他の重要な価値                   | 123        |
| 18          | AIガバナンス                     | 124        |
| <b>巻末</b>   | <b>総索引</b>                  | <b>128</b> |

## 第1部

# 人工知能とは

技術分野／2の中項目・18語

- 1 人工知能の定義（5語）
- 2 人工知能分野で議論される問題（13語）

## 1 人工知能の定義

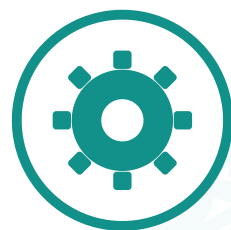
⇒ **この節の目標。**人工知能とは何か、具体例を用いて／人工知能のレベルを「単純な制御プログラム」「古典的な人工知能」「機械学習」「深層学習」の4つに分類し、それぞれがどのようなものか説明することができる／AI効果／人工知能とロボットの違いを説明できる、人工知能とは何か、具体例を用いて。

### AI効果

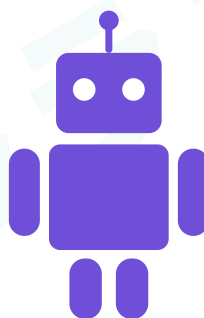
AIによって何かが実現されると、その仕組みが分かった途端に「これは単なる自動化であって知能ではない」と受け取られるようになる現象です。人工知能の成果は、達成されるたびに人工知能の枠から外されていきます。そのためAIは、いつまでも「まだできていないもの」に見え続けます。

### エージェント

周囲の状況を認識し、自分で判断して行動するプログラムを指します。環境を知覚し、次の一手を決め、実際に働きかけ、その結果をまた知覚する。この繰り返しで目標に近づいていきます。人工知能を「環境の中で目標を達成するもの」としてとらえる見方の中心にあり、のちに学ぶ強化学習も同じ枠組みで説明されます。



**AI = ソフトウェア**  
考える中身（頭）



**ロボット = ハードウェア**  
体

図 エージェント

### 人工知能

人間の知的な働きをコンピュータで実現しようとする技術と、その研究分野のことです。注意したいのは、専門家の間で合意された定義が、いまでも存在しないという点です。研究者ごとに「推論する機械」「人間のように振る舞う機械」など、とらえ方が分かれたままになっています。



図 人工知能

### 機械学習

人がルールを書き下す代わりに、大量のデータからパターンを学び取らせる方法です。たとえば迷惑メールの判定では、条件を人が書くのではなく、過去のメールから見分け方そのものを学ばせます。データが増えるほど精度が上がるのが期待でき、ルールを書ききれない問題に強いのが特長です。

### ディープラーニング

層を深く重ねたニューラルネットワークを使う、機械学習の一つの方法です。それまでの機械学習では、どこに着目するかという特徴を人が設計していました。ディープラーニングは、その特徴そのものをデータから自動的に学びます。この違いが、画像や音声の認識精度を大きく押し上げました。



図 ディープラーニング

## 2 人工知能分野で議論される問題

→ **この節の目標。**人工知能分野で議論されている代表的な問題について／汎用的な人工知能の実現可能性について、いくつかの例を取り上げて。

### シンギュラリティ

技術的特異点と訳されます。AIが自分より賢いAIを作れるようになり、その連鎖によって、人間には予測できない速さで進歩が進むとされる時点のことです。レイ・カーツワイルは、これを2045年と予測しました。あくまで予測であり、到来するかどうかは研究者の間でも意見が分かれています。

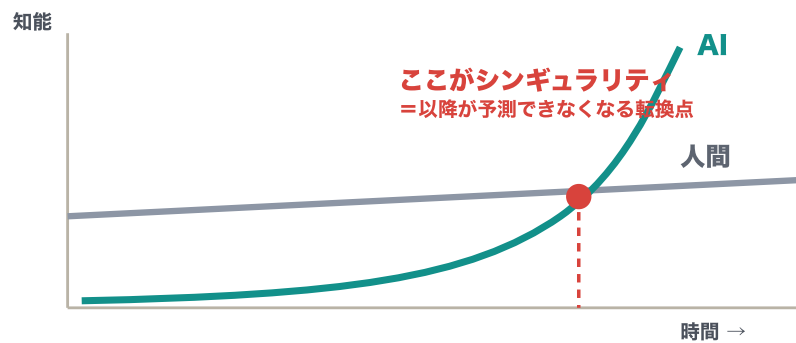


図 シンギュラリティ

## シンボルグラウンディング問題

記号と、それが指す実世界の意味を、どう結びつけるかという問題です。たとえばコンピュータが「シマウマ」という言葉と「縞模様の馬」という説明を知っていても、初めて見た実物をシマウマだと結びつけられるとは限りません。記号を扱えることと、意味が分かっていることは別だ、という指摘です。

## 身体性

知能は、身体を通して世界と関わることで初めて成り立つ、という考え方です。人間は、触ったり動いたりした経験と結びつけて言葉の意味をつかんでいます。ロドニー・ブルックスは、記号を操作することより先に、環境と直接やりとりする仕組みが必要だと主張しました。シンボルグラウンディング問題に対する一つの答えでもあります。

## ダートマス会議

1956年に、アメリカのダートマス大学で開かれた研究会です。ジョン・マッカーシーが、この会議の提案の中で「人工知能」という言葉を初めて使いました。マービン・ミンスキーやクロード・シャノンらも参加しています。人工知能という分野が始まった場として、試験でもよく問われます。

## トイ・プロブレム

迷路や積み木、パズルのように、規則が単純で、扱う範囲が限られた問題のことです。第一次AIブームのころのAIは、こうした問題なら解くことができました。しかし現実の問題は条件が多く曖昧で、同じやり方では歯が立ちません。ブームが冷え込んだ原因として挙げられます。

## 知識獲得のボトルネック

専門家の知識をコンピュータに書き移す作業に、非常に大きな手間がかかることを指します。第二次AIブームのエキスパートシステムは、知識を人手でルールとして入力する必要がありました。分野が広がるほど作業量が増え、更新も追いつかず、これが普及を妨げる壁になりました。

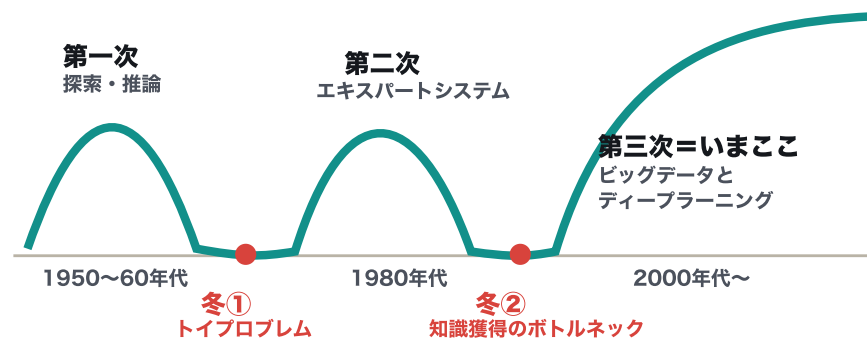


図 知識獲得のボトルネック

## チューリングテスト

機械が知能を持つといえるかどうかを判定する方法です。アラン・チューリングが1950年に提案しました。判定者が文字のやり取りだけで会話し、相手が機械か人間かを見分けられなければ、その機械は知能を持つとみなします。中身の仕組みではなく、外から見た振る舞いで判断するのが要点です。

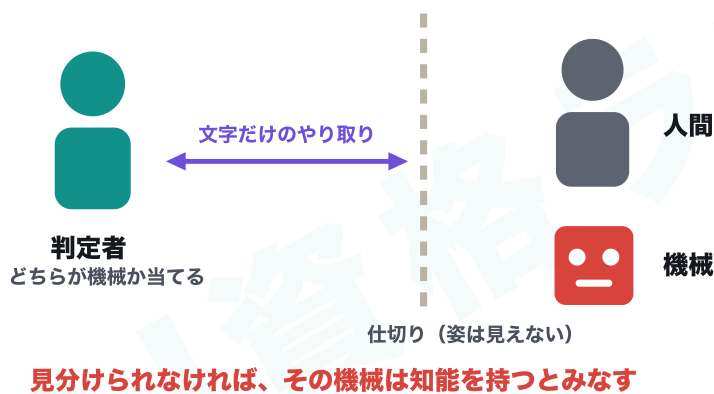


図 チューリングテスト

## 中国語の部屋

ジョン・サールが示した思考実験です。部屋の中の人々が、マニュアルに従って中国語の文字を並べ替えて返すと、外からは中国語を理解しているように見えます。しかし本人は中国語を全く分かっていません。記号を正しく処理できることと、意味を理解していることは別だ、という反論です。

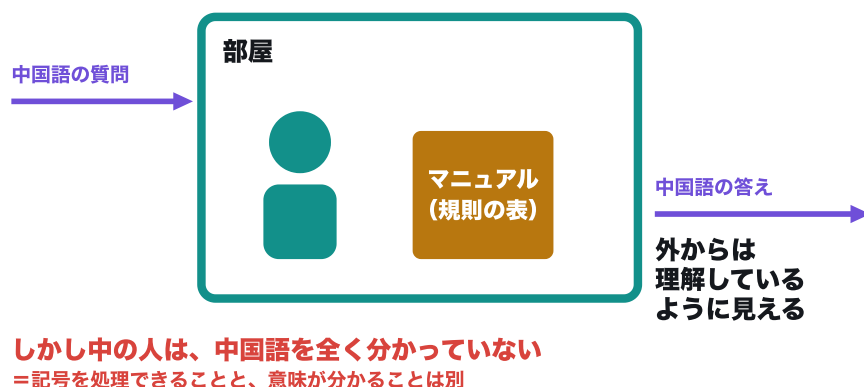


図 中国語の部屋

## 強いAIと弱いAI

ジョン・サールによる区別です。強いAIは、人間と同じように心や理解そのものを持つAIを指します。弱いAIは、心は持たないものの、特定の作業を道具としてこなすAIです。いま実用になっているものは、すべて弱いAIに当たります。中国語の部屋は、強いAIの立場に対する反論として示されました。

## 統計的機械翻訳

大量の対訳データから、どの訳し方がどれくらい確からしいかを統計的に学んで翻訳する方式です。人が文法規則を書く方式に代わって主流になり、規則を書ききれないという弱点を乗り越えました。その後、ニューラルネットワークを使うニューラル機械翻訳に置き換わっています。

## フレーム問題

ある行動をするときに、関係のあることだけを取り出して考えるのが難しい、という問題です。ジョン・マッカーシーらが指摘しました。起こりうることをすべて検討していると計算が終わらず、いつまでも行動できません。人間が無意識にやっている取舍選択を機械にさせる難しさを表しています。

## ルールベース機械翻訳

文法の規則と辞書を人が書き、そのとおりに訳す方式です。統計的機械翻訳より前の主流でした。規則がはっきりしているぶん、なぜその訳になったのかを説明できます。一方で、言語ごとに規則を作り込む手間が非常に大きく、例外にも弱いという問題を抱えていました。

## ローブナーコンテスト

チューリングテストにもとづいて、最も人間らしい対話ができるプログラムを選ぶコンテストです。1991年から開催されてきました。実際に判定者と会話させて順位を付けるもので、チューリングテストを現実に実施した例として知られています。

## 確認問題 第1部のふりかえり

### 確認問題 第1部

#### 第1問 人工知能における「エージェント」の説明として、最も適切な選択肢を1つ選べ

答

**環境を認識し、自ら判断して行動する主体**

エージェントは、環境を知覚し、判断し、働きかけ、その結果をまた知覚する、という繰り返りで目標に近づく主体を指します。強化学習でも同じ枠組みが使われます。

### 確認問題 第1部

#### 第2問 機械学習とディープラーニングの違いとして、最も適切な選択肢を1つ選べ

答

**ディープラーニングは、着目すべき特徴そのものをデータから自動で学ぶ**

従来の機械学習では、どこに着目するかという特徴を人が設計していました。ディープラーニングはその特徴そのものを学習で獲得します。この違いが精度を大きく押し上げました。

### 確認問題 第1部

#### 第3問 ジョン・サールが示した「中国語の部屋」の主張として、最も適切な選択肢を1つ選べ

答

**記号を正しく処理できることと、意味を理解していることは別である**

部屋の中の人マニュアルに従って中国語を返しますが、中国語を理解していません。記号処理ができることは理解していることを意味しない、という強いAIへの反論です。

## 18語チェックリスト

自分の言葉で説明できたら✓を入れてください。読めるだけでは足りません。

- |                                    |                                         |                                     |
|------------------------------------|-----------------------------------------|-------------------------------------|
| <input type="checkbox"/> AI効果      | <input type="checkbox"/> シンボルグラウンディング問題 | <input type="checkbox"/> 中国語の部屋     |
| <input type="checkbox"/> エージェント    | <input type="checkbox"/> 身体性            | <input type="checkbox"/> 強いAIと弱いAI  |
| <input type="checkbox"/> 人工知能      | <input type="checkbox"/> ダートマス会議        | <input type="checkbox"/> 統計的機械翻訳    |
| <input type="checkbox"/> 機械学習      | <input type="checkbox"/> トイ・プロブレム       | <input type="checkbox"/> フレーム問題     |
| <input type="checkbox"/> ディープラーニング | <input type="checkbox"/> 知識獲得のボトルネック    | <input type="checkbox"/> ルールベース機械翻訳 |
| <input type="checkbox"/> シンギュラリティ  | <input type="checkbox"/> チューリングテスト      | <input type="checkbox"/> ロープナーコンテスト |

## 第2部

# 人工知能をめぐる動向

技術分野／4の中項目・36語

- 3 探索・推論（10語）
- 4 知識表現とエキスパートシステム（14語）
- 5 機械学習（5語）
- 6 ディープラーニング（7語）

### 3 探索・推論

⇒ **この節の目標。**探索・推論の具体例／探索・推論を実現する代表的な手法。

#### $\alpha\beta$ 法

Mini-Max法の読みを効率よくする方法です。途中まで読んだ時点で、これ以上調べても結論が変わらないと分かった枝を切り捨てます。自分の手番で切ることを $\alpha$ カット、相手の手番で切ることを $\beta$ カットと呼びます。読む量を減らしても、選ぶ手は変わりません。

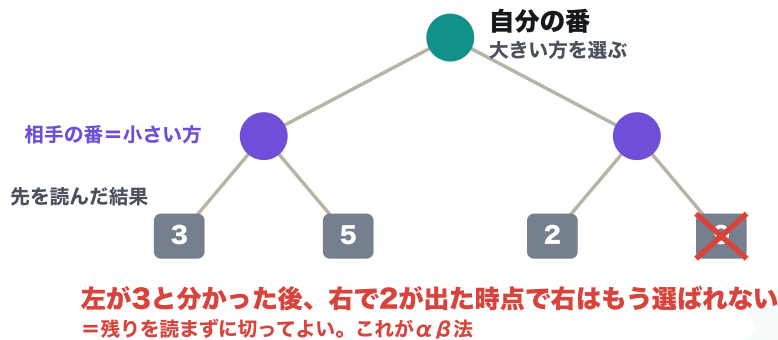


図  $\alpha\beta$ 法

#### Mini-Max法

二人で交互に指すゲームで先を読むための方法です。自分の手番では自分にとって最も良い手を、相手の手番では相手が自分にとって最も悪い手を選ぶ、と仮定して読み進めます。ゲーム木をたどって、最後に一番ましな結果になる手を選びます。

#### SHRDLU

テリー・ウィノグラードが1970年前後に作った、積み木の世界を扱うシステムです。「赤い積み木を青い箱に入れて」といった自然言語の指示を理解して動かせました。ただし積み木という限られた世界だけで動くもので、現実の世界には広がりませんでした。

#### STRIPS

目標に至る行動の手順を組み立てる、プランニングの手法です。1971年にリチャード・ファイクスとニルス・ニルソンが発表しました。それぞれの行動を「前提条件」「削除リスト」「追加リスト」の3つで書き表し、目標の状態に近づく順番を探します。

#### 探索木

取りうる場合を木の形に並べ、枝をたどって答えを探す考え方です。根が最初の状態、枝分かれが選べる手、葉が結果に当たります。探索という手法の土台であり、幅優先探索も深さ優先探索も、この木をどの順にたどるかの違いです。

## ハノイの塔

3本の棒と大きさの違う円盤を使うパズルです。円盤を1枚ずつ動かし、大きい円盤を小さい円盤の上に置かないという規則で、全部を別の棒へ移します。規則が単純で探索で解けるため、初期のAI研究で使われた代表的なトイ・プロブレムです。

## 幅優先探索

探索木を、根に近い階層から順に、横方向へすべて調べていく方法です。最も浅い所にある答え、つまり最短の手順を確実に見つけられます。一方で、その階層の候補をすべて覚えておく必要があるため、メモリを多く使うという弱点があります。

## 深さ優先探索

探索木を、行けるところまで一気に下り、行き止まりに当たったら一つ戻って別の枝へ進む方法です。覚えておく量が少なくて済むのが利点です。一方で、見つかった答えが最短とは限らず、枝が深いと戻ってこれなくなることもあります。

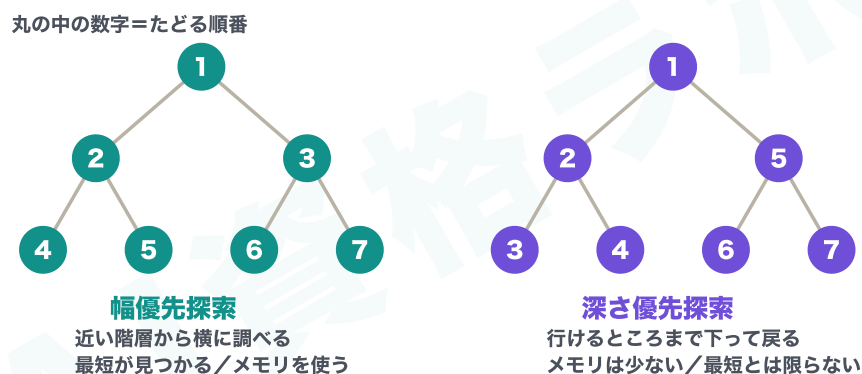


図 深さ優先探索

## ブルートフォース

力任せの探索とも呼ばれ、すべての場合をしらみつぶしに調べる方法です。答えがあれば必ず見つかるという確実さがあります。しかし場合の数は手数が増えるほど爆発的に増えるため、囲碁や将棋のような組み合わせの多い問題では、現実的な時間で終わりません。

## モンテカルロ法

乱数を使って何度も試し、その結果から良い手を見積もる方法です。最後まで読み切る代わりに、ランダムに終局まで打ってみて勝率の高い手を選びます。すべてを読み切れない囲碁で特に有効で、アルファ碁にも取り入れられました。

## 4 知識表現とエキスパートシステム

→ **この節の目標。** 知識表現とは何か／エキスパートシステムとは何か／知識表現に係る代表的な研究や手法について／エキスパートシステムに係る代表的な研究や手法について。

## Cycプロジェクト

人間が持つ一般常識を、すべてコンピュータに書き込もうという試みです。1984年にダグラス・レナートが始め、現在も続いています。膨大な知識を人手で入れ続ける必要があり、知識獲得のボトルネックがどれほど大きいかを示す例として挙げられます。

## DENDRAL

1960年代にスタンフォード大学で作られた、世界初のエキスパートシステムとされるプログラムです。質量分析のデータから、その物質の化学構造を推定します。専門家の知識をルールとして持たせれば実用になる、と示した点で画期的でした。

## is-aの関係・has-aの関係・part-ofの関係

意味ネットワークで、概念どうしを結ぶ代表的な関係です。is-aは「～の一種である」という上下の関係、has-aは「～を持っている」という所有の関係、part-ofは「～の一部である」という包含の関係を表します。

## Question-Answering

質問応答と訳されます。検索エンジンのように関係しそうな文書を並べるのではなく、質問に対する答えそのものを返す仕組みです。IBMのワトソンが代表例で、大量の文書から答えの候補を集め、確からしさで順位を付けます。いまの対話型AIにもつながる考え方です。

## 意味ネットワーク

知識を書き表す方法の一つです。概念を点、概念どうしの関係を線で表し、網の形にして整理します。もとは人間の記憶の仕組みを説明するために考えられたもので、is-aやpart-ofといった関係で結ぶことで、コンピュータが知識をたどれるようにします。

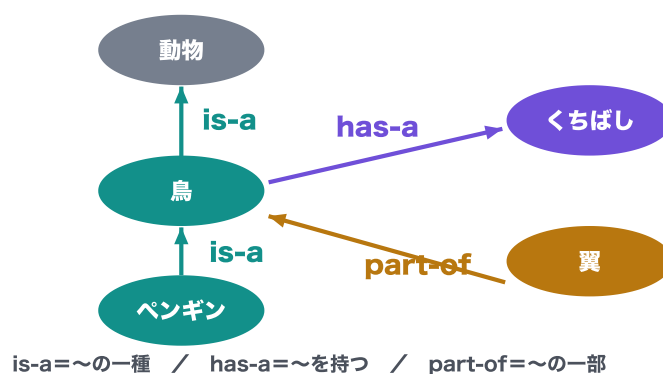


図 意味ネットワーク

## イライザ ELIZA

1960年代にジョセフ・ワイゼンバウムが作った対話プログラムです。相手の言葉を型に当てはめて言い換えて返すだけの仕組みでしたが、本物のカウンセラーと話していると錯覚する人が現れました。これをイライザ効果と呼びます。

## インタビューシステム

専門家に質問を重ねることで知識を引き出し、ルール of 形で取り込む仕組みです。エキスパートシステムを作るとき、知識を書き移す作業に非常に手間がかかりました。その負担を減らすための工夫であり、知識獲得のボトルネックへの対策の一つです。

## ウェブマイニング

ウェブ上のページの内容や、利用者の閲覧履歴から、役に立つ知識を取り出すことです。データマイニングをウェブに当てはめたもので、どのページがよく参照されるか、どの語が一緒に現れるか、といった規則を見つけます。検索の順位付けやおすすめの実現に使われます。

## オントロジー

概念と概念どうしの関係を、誰が使っても同じ意味になるようにきちんと定義したものです。もとは哲学の存在論を指す言葉でした。人手で厳密に作るヘビーウェイトオントロジーと、機械で自動的に作るライトウェイトオントロジーに分けられます。

## セマンティックWeb

ウェブ上の情報に意味を表す情報を付け、コンピュータが内容そのものを扱えるようにしようという構想です。ウェブの生みの親であるティム・バーナーズ＝リーが提唱しました。オントロジーは、この構想を支える技術として位置づけられます。

## データマイニング

大量のデータの中から、それまで気づかれていなかった規則やパターンを掘り出すことです。鉱山で鉱石を掘り出すことにたとえた言葉です。専門家の知識を人が書き入れるエキスパートシステムとは逆に、知識をデータの側から見つけにいく点が、次の機械学習へつながる流れになります。

## 東ロボくん

国立情報学研究所が2011年から進めた、東京大学の入試合格を目指すプロジェクトです。多くの科目で成績を伸ばしましたが、文章の意味を読み取る力の壁を越えられず、2016年に開発が凍結されました。読解の難しさを示した例としてよく挙げられます。

## マイシン MYCIN

1970年代にスタンフォード大学で作られたエキスパートシステムです。血液の細菌感染症を診断し、使うべき抗生物質を提案しました。専門医に近い正答率を出した一方で、責任の所在という問題もあり、実際の医療現場では使われませんでした。

## ワトソン

IBMが開発した質問応答システムです。2011年にアメリカのクイズ番組ジェパディ！に出場し、人間のチャンピオンに勝ちました。膨大な文書を検索して根拠を集め、答えの確からしさを点数にして選ぶ仕組みで、意味を理解しているわけではありません。

## 5 機械学習

⇒ **この節の目標。**機械学習とは何か／機械学習とルールベース手法の差異、およびメリットデメリットについて／機械学習が注目されるようになった背景／機械学習がどのような場面において、効果を発揮するのか／機械学習の代表的な応用例について。

### 次元の呪い

扱う特徴の数、つまり次元が増えるほど、学習に必要なデータが爆発的に増えてしまうことです。次元が高くなるとデータどうしの距離の差が小さくなり、似ているかどうかの判断もつきにくくなります。特徴を絞り込む工夫が必要になる理由です。

### スパムフィルタ

迷惑メールを自動で見分ける仕組みです。人が条件を書くのではなく、過去に届いたメールから見分け方そのものを学びます。新しい手口が出てきても学び直せば対応でき、機械学習が力を発揮する場面の代表例としてよく挙げられます。

### ビッグデータ

量が多く、種類も多様で、発生する速度も速いデータのことです。インターネットとセンサーの普及によって集まるようになりました。機械学習はデータが多いほど精度が上がるため、ビッグデータの登場が、機械学習が実用になった大きな背景です。

### レコメンデーションエンジン

利用者の行動や履歴から好みを推定し、商品や動画をすすめる仕組みです。似た人が好んだものをすすめる方法と、対象そのものの特徴が似ているものをすすめる方法があります。機械学習が事業に直接結びついた、分かりやすい応用例です。

## 統計的自然言語処理

言葉を統計的に扱い、どの解釈がどれくらい確からしいかで処理する方法です。人が文法の規則を書き下す方式では、例外や曖昧さに対応しきれませんでした。大量の文章から確からしさを学ぶこの方式が主流となり、翻訳や音声認識を大きく進めました。

## 6 ディープラーニング

⇒ **この節の目標。**ディープラーニングがどのように発展してきたのか、その歴史／古典的な機械学習とディープラーニングの差異／ディープラーニングの代表的な応用例について。

### ImageNet

1400万枚を超える画像に、何が写っているかのラベルを付けた大規模なデータセットです。スタンフォード大学のフェイフェイ・リーらが構築しました。学習に使える大量のデータがそろったことが、ディープラーニングが力を示す土台になりました。

### ILSVRC

ImageNetを使って画像認識の精度を競うコンテストです。2012年にトロント大学のチームがディープラーニングを用いたAlexNetで、それまでの手法に大差をつけて優勝しました。これが第三次AIブームの引き金になったとされます。

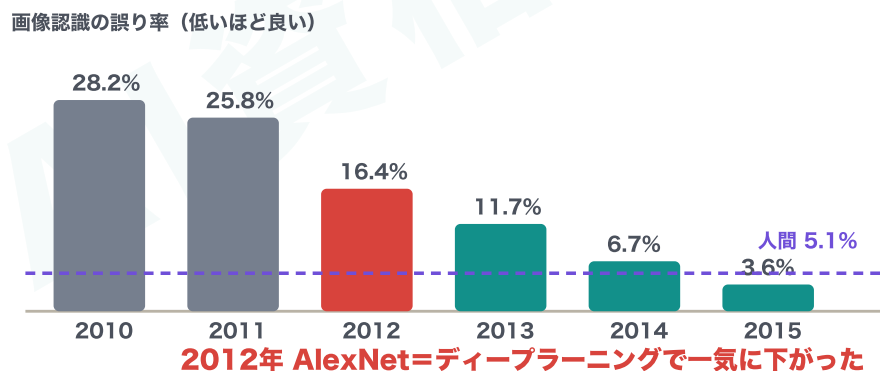


図 ILSVRC

### LeNet

1998年にヤン・ルカンが発表した、畳み込みニューラルネットワークです。手書きの数字を認識するために作られ、郵便番号の読み取りなどに使われました。畳み込み層とプーリング層を重ねるという、いまも使われる基本の形をすでに備えていました。

### アルファ碁 AlphaGo

DeepMindが開発した囲碁のプログラムです。2016年に世界トップ級の棋士イ・セドルに勝ちました。ディープラーニングで有望な手と勝率を見積もり、モンテカルロ木探索で読み、さらに自己対戦による強化学習で強くするという組み合わせが特徴です。

## 人間の神経回路

ニューラルネットワークが手本にした、脳の神経細胞のつながりのことです。神経細胞は、他の細胞から受け取った信号の合計がある大きさを超えると、次へ信号を伝えます。この「重みを付けて足し合わせ、しきい値で発火する」仕組みを数式にしたものが出発点です。

## ネオコグニトロン

1979年に福島邦彦が提案した、人間の視覚のしくみをまねた多層のモデルです。単純な特徴を捉える層と、位置のずれを吸収する層を交互に重ねる構造でした。これは、のちの畳み込みニューラルネットワークの源流にあたります。

## 生成AI

文章や画像、音声などを新しく作り出すAIの総称です。大量のデータで学んだモデルが、学んだ内容をもとに、これまでになかったものを出力します。ディープラーニングの発展の先に現れ、2022年ごろから急速に広まって、いまのAIの中心になっています。

## 確認問題 第2部のふりかえり

### 確認問題 第2部

#### 第1問 深さ優先探索の説明として、最も不適切な選択肢を1つ選べ

答

**常に最も少ない手数の解を見つけられる**

深さ優先探索は最初に見つかった解を返すため、最短とは限りません。最短を保証するのは幅優先探索です。

### 確認問題 第2部

#### 第2問 $\alpha$ $\beta$ 法の説明として、最も適切な選択肢を1つ選べ

答

**読んでも結論が変わらないと分かった枝を、途中で切り捨てる**

$\alpha$   $\beta$  法はMini-Max法の読みを効率化する方法です。これ以上調べても選ぶ手が変わらないと分かった枝を切り捨てます。読む量が減っても、選ぶ手は変わりません。

### 確認問題 第2部

#### 第3問 以下の文章を読み、空欄に最もよく当てはまる選択肢を1つ選べ 二人で交互に指すゲームで、自分の手番では最も良い手を、相手の手番では自分にとって最も悪い手が選ばれると仮定して読み進める方法を（ ）という

答

**Mini-Max法**

Mini-Max法は、自分は最大、相手は最小になるよう選ぶと仮定してゲーム木をたどり、最後に最もましな結果になる手を選びます。

## 36語チェックリスト

自分の言葉で説明できたら✓を入れてください。読めるだけでは足りません。

- |                                             |                                                      |                                      |
|---------------------------------------------|------------------------------------------------------|--------------------------------------|
| <input type="checkbox"/> $\alpha$ $\beta$ 法 | <input type="checkbox"/> モンテカルロ法                     | <input type="checkbox"/> オントロジー      |
| <input type="checkbox"/> Mini-Max法          | <input type="checkbox"/> Cycプロジェクト                   | <input type="checkbox"/> セマンティックWeb  |
| <input type="checkbox"/> SHRDLU             | <input type="checkbox"/> DENDRAL                     | <input type="checkbox"/> データマイニング    |
| <input type="checkbox"/> STRIPS             | <input type="checkbox"/> is-aの関係・has-aの関係・part-ofの関係 | <input type="checkbox"/> 東ロボくん       |
| <input type="checkbox"/> 探索木                | <input type="checkbox"/> Question-Answering          | <input type="checkbox"/> マイシン(MYCIN) |
| <input type="checkbox"/> ハノイの塔              | <input type="checkbox"/> 意味ネットワーク                    | <input type="checkbox"/> ワトソン        |
| <input type="checkbox"/> 幅優先探索              | <input type="checkbox"/> イライザ(ELIZA)                 | <input type="checkbox"/> 次元の呪い       |
| <input type="checkbox"/> 深さ優先探索             | <input type="checkbox"/> インタビューシステム                  | <input type="checkbox"/> スパムフィルタ     |
| <input type="checkbox"/> ブルートフォース           | <input type="checkbox"/> ウェブマイニング                    | <input type="checkbox"/> ビッグデータ      |

- ☐ レコメンデーションエンジン
- ☐ 統計的自然言語処理
- ☐ ImageNet

- ☐ ILSVRC
- ☐ LeNet
- ☐ アルファ碁(AlphaGo)

- ☐ 人間の神経回路
- ☐ ネオコグニトロン
- ☐ 生成AI

AI資格ラボ

### 第3部

# 機械学習の概要

技術分野／4の中項目・62語

- 7 教師あり学習 (20語)
- 8 教師なし学習 (14語)
- 9 強化学習 (12語)
- 10 モデルの選択・評価 (16語)

## 7 教師あり学習

→ **この節の目標。**教師あり学習には、特徴量と教師データのペアが必要であること／教師あり学習における、分析対象に応じた問題の種類を列挙・／代表的な教師あり学習モデルの基本概念／目的やデータの特性・量に応じて、適切な教師あり学習モデルを選択できる／ビジネスにおける教師あり学習の応用例。

### アンサンブル学習

複数のモデルを組み合わせて、一つのモデルより良い結果を得ようとする方法です。一つひとつは弱くても、多数決や平均を取ることで誤りが打ち消し合います。代表的なやり方に、バギング、ブースティング、スタッキングがあります。

### カーネル

データをより高い次元へ写す働きをする関数です。もとの空間では直線で分けられないデータでも、高い次元へ写すと平面で分けられることがあります。サポートベクターマシンを非線形の問題に使えるようにする鍵になります。

### カーネルトリック

高い次元へ実際にデータを写す計算をせずに、写した先での内積だけをカーネル関数で求めてしまう工夫です。次元が上がるほど計算量は爆発しますが、この工夫によって現実的な時間で扱えるようになります。

### 回帰問題

連続する数値を予測する問題です。売上や気温、価格などが対象になります。どのグループに属するかを当てる分類問題と並ぶ、教師あり学習の二大分類の一つです。評価には平均二乗誤差などの、誤差の大きさを測る指標を使います。

### 決定木

条件による枝分かれを繰り返して予測する、木の形をしたモデルです。「年齢が30以上か」といった問いを重ねて答えにたどり着きます。なぜその答えになったかを人がたどれる点が大きな長所で、説明が求められる場面で使われます。

### 勾配ブースティング

前のモデルが出した誤差を、次のモデルが埋めるように順番に学習させる、ブースティングの一種です。誤差の減る方向、つまり勾配にしたがって重ねていきます。XGBoostやLightGBMとして広く使われ、表形式のデータでは今も強力です。

## サポートベクターマシン SVM

データを分ける境界を、最も近いデータとの余白が最大になるように引く手法です。この考え方をマージン最大化と呼びます。カーネルトリックと組み合わせることで非線形にも対応でき、ディープラーニング以前の主力の一つでした。

## 線形回帰

データの間関係を直線、変数が増えれば平面や超平面で表す、最も基本的な回帰の手法です。予測値と実際の値の差の二乗和が最小になるように直線を決めます。単純ですが、結果の解釈がしやすいという利点があります。

## 自己回帰モデル(ARモデル)

過去の自分自身の値から次の値を予測する、時系列データのモデルです。今日の気温を、昨日や一昨日の気温から予測するような形になります。時系列を扱う手法の出発点として押さえておく必要があります。

## 単回帰分析

説明変数が一つだけの回帰分析です。たとえば気温だけからアイスの売上を予測する場合があります。関係を一本の直線で表すため、グラフに描いて確かめやすいのが利点です。説明変数が複数ある重回帰分析と対にして覚えます。

## 重回帰分析

説明変数が複数ある回帰分析です。気温、曜日、天気などを同時に使って売上を予測します。変数どうしの相関が強すぎると、係数が不安定になって解釈できなくなる多重共線性という問題が起きるため、注意が必要です。

## 多クラス分類

三つ以上のグループのどれに当たるかを当てる分類です。手書き数字を0から9のどれかに分ける場合が当たります。二つに分ける二値分類と対比して問われます。出力を確率に直すソフトマックス関数がよく使われます。

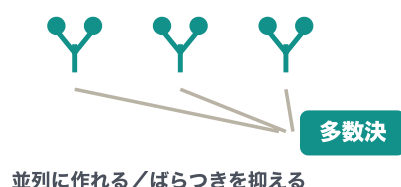
## バギング

学習データから重複を許して選び直したものでモデルを複数作り、多数決や平均を取る方法です。それぞれ独立に学習できるので並列に処理でき、予測のばらつきを抑える効果があります。ランダムフォレストがこの代表です。

## ブースティング

前のモデルが間違えたデータを重く見ながら、モデルを順番に作っていく方法です。誤りを一つずつ潰していくため精度は上がりやすい一方、前のモデルが終わらないと次に進めないで、並列化できず時間がかかります。

### バギング=同時に作って多数決



### ブースティング=順番に作る

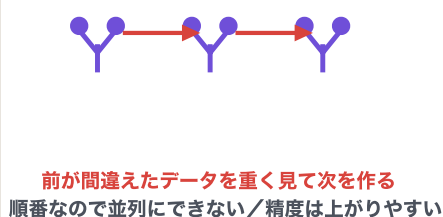


図 ブースティング

## ブートストラップサンプリング

もとのデータから重複を許して、同じ数だけ選び直すことです。選ばれないデータも出るため、モデルごとに少しずつ違う学習データができます。この違いがバギングの効果を生む土台になっています。

## 分類問題

データがどのグループに属するかを当てる問題です。迷惑メールかどうか、どの数字が書かれているか、といった形になります。連続する数値を予測する回帰問題と対になり、評価には正解率や適合率などを使います。

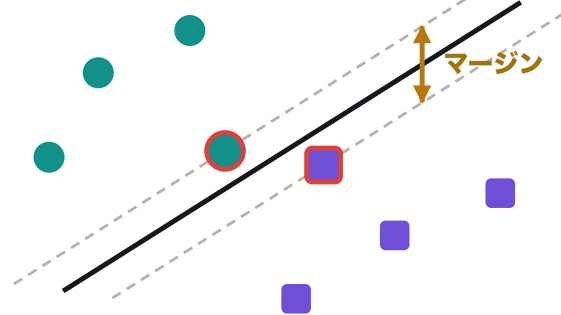
## ベクトル自己回帰モデル(VARモデル)

複数の時系列が互いに影響し合う形に広げた自己回帰モデルです。金利と株価のように、一方の過去の値がもう一方の将来に効く関係を同時に扱えます。単独の系列を扱うARモデルの拡張として押さえます。

## マージン最大化

分ける境界と、それに最も近いデータとの距離を最大にするという考え方です。この距離をマージンと呼びます。余白を大きく取ることで、初めて見るデータにも当てはまりやすくなります。サポートベクターマシンの中心にある考え方です。

この余白（マージン）が最大になるように境界を引く



**境界**  
2つの破線のまん中に引く

**サポートベクター**  
境界に最も近いデータ  
=破線の上に乗っている点

図 マージン最大化

## ランダムフォレスト

決定木をバギングで多数作り、さらに各木が使う特徴もランダムに選ぶ手法です。木どうしが似すぎるのを防ぐことで、組み合わせた効果がより大きくなります。調整すべき設定が少なく扱いやすいため、長く広く使われてきました。

## ロジスティック回帰

名前に回帰と付きますが、分類に使う手法です。線形回帰の出力をシグモイド関数に通して0から1の値にし、それを確率とみなして分けます。結果を確率として解釈できるため、医療や与信の分野でよく使われます。

## 8 教師なし学習

⇒ **この節の目標。**教師なし学習には、特徴量のみが必要であること／教師なし学習における、分析対象に応じた問題の種類を列挙・／代表的な教師なし学習モデルの基本概念／目的やデータの特性・量に応じて、適切な教師なし学習モデルを選択できる／ビジネスにおける教師なし学習の応用例。

### k-means法

データをk個のかたまりに分ける、代表的な非階層クラスタリングです。まず重心をk個置き、各データを近い重心に割り当て、割り当てた結果から重心を計算し直す、という手順を繰り返します。kの数はあらかじめ人が決めます。

### t-SNE

高い次元のデータを二次元などに落として、目で見えるようにする次元削減の手法です。もとの空間で近いものが、落とした先でも近くに来るように配置します。クラスタの様子を確かめる可視化の用途でよく使われます。

## ワード法

階層クラスタリングの代表的な方法です。二つのかたまりを一つにまとめたときに、内部のばらつきの増え方が最も小さくなる組を選んで順に併合していきます。まとまりの良いクラスタがしやすいという特徴があります。

## 協調フィルタリング

自分と似た好みを持つ他の利用者の行動をもとに、商品や動画をすすめる方法です。対象そのものの自身は見ず、評価の履歴だけを使います。そのため履歴のない新しい利用者や商品には使えず、コールドスタート問題を抱えます。

## クラスタリング

似たもの同士をグループにまとめることです。正解ラベルを与えずに構造を見つけるので、教師なし学習に当たります。木の形にまとめる階層クラスタリングと、あらかじめ数を決める非階層クラスタリングに分かれます。

## コールドスタート問題

新しい利用者や新しい商品は履歴がないため、協調フィルタリングでは推薦ができない、という問題です。対策として、対象そのものの特徴を使うコンテンツベースフィルタリングと組み合わせる方法がよく取られます。

## コンテンツベースフィルタリング

すすめる対象そのものの特徴が似ているものを推薦する方法です。同じジャンルや同じ作者、といった情報を使います。他人の履歴を必要としないため、コールドスタート問題に強いという利点があります。

## 次元削減

扱う特徴の数を減らして、データを小さく扱いやすくすることです。特徴が増えるほど学習が難しくなる次元の呪いへの対策であり、二次元に落として目で見えるための可視化にも使われます。主成分分析やt-SNEが代表です。

## 主成分分析 PCA

データの散らばりが最も大きい方向を新しい軸として取り直し、次元を減らす代表的な手法です。情報の失われ方を小さく抑えながら、特徴の数を減らせます。教師なし学習に分類され、可視化にも前処理にも使われます。

## 潜在的ディリクレ配分法 LDA

一つの文書が複数の話題の混ざり合いでできている、と考えるトピックモデルの代表的な手法です。どの語がどの話題に現れやすいかを推定し、文書ごとの話題の割合を求めます。文書の分類や整理に使われます。

## 多次元尺度構成法 MDS

データどうしの距離の関係をできるだけ保ったまま、低い次元に配置し直す手法です。もとの特徴が分からなくても、距離さえ分かれば使えるのが特徴で、似ているものの位置関係を目で見たいときに用います。

## デンドログラム(樹形図)

階層クラスタリングの結果を木の形で表した図です。どのデータがどの順に併合されたかが縦の高さで分かります。どの高さで横に切るかによってクラスタの数が決まるため、切る位置を人が判断できるのが利点です。

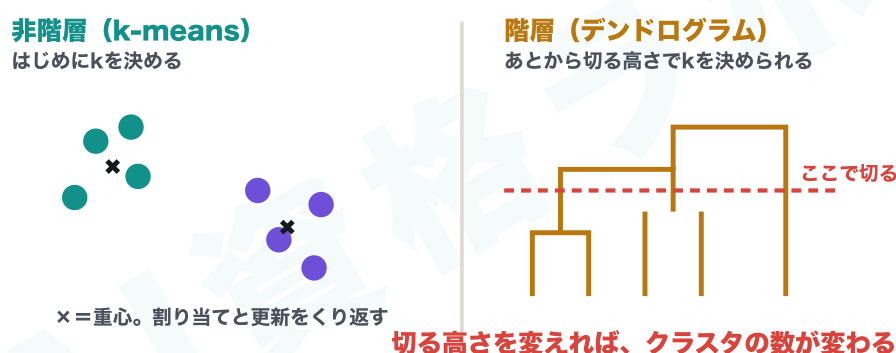


図 デンドログラム(樹形図)

## 特異値分解 SVD

一つの行列を三つの行列の積に分解する方法です。分解したうち重要な成分だけを残せば、もとの情報をおおよそ保ったまま次元を減らせます。利用者と商品の評価をまとめた行列に当てはめて、推薦に使うこともあります。

## トピックモデル

文書の背後にある話題を推定する、教師なし学習のモデルです。あらかじめ分類先を決めなくても、文書の集まりからどんな話題があるかを見つけ出せます。代表的な手法が潜在的ディリクレ配分法です。

# 9 強化学習

⇒ **この節の目標。** 強化学習の基本概念／強化学習と、教師あり学習および教師なし学習との差異／価値関数の学習と、方策の学習の2つの代表的なアプローチ／各アプローチに属する代表的な強化学習手法について概要／ビジネスにおける強化学習の応用例。

## Actor-Critic

行動を決める役割のActorと、その良さを見積もる役割のCriticを組み合わせる強化学習の方法です。方策を直接改善する方策勾配法と、価値を見積もる価値ベースの手法の、両方の長所を取り入れています。

## $\epsilon$ -greedy方策

ほとんどの場合は今いちばん良いとされる行動を選び、ある確率 $\epsilon$ だけ、あえて別の行動を試す方策です。良いと分かっているものを使う活用と、まだ試していないものを調べる探索の、釣り合いを取るための最も基本的な工夫です。

## REINFORCE

方策勾配法の代表的な手法です。実際に行動してみて得られた報酬をもとに、良い結果につながった行動が選ばれやすくなるよう、方策を直接更新します。Actor-Criticの土台にもなっています。

## Q学習

ある状態である行動を取ったときの価値、つまりQ値を、経験しながら更新していく手法です。実際にはどの方策で動いたかにかかわらず、最も価値の高い行動を前提に更新するため、方策オフ型と呼ばれます。SARSAとの違いがよく問われます。

## UCB方策

これまでの平均的な良さに加えて、試した回数の少なさも足し合わせて選ぶ方策です。あまり試していない選択肢ほど加点されるので、自然に探索が促されます。バンディット問題で使われる代表的な方策です。

## 行動価値関数

ある状態で、ある行動を取ったときに、その後で得られる報酬の見込みを表す関数です。Q値とも呼ばれ、Q学習という名前もここから来ています。状態だけでなく行動まで含めて評価する点が、状態価値関数との違いです。

## 状態価値関数

ある状態にること自体の良さ、つまりその状態から先で得られると見込まれる報酬を表す関数です。どの行動を取るかまでは含めません。行動まで含めて評価する行動価値関数と、対にして違いを押さえます。

## バンディットアルゴリズム

複数の選択肢のうちどれを選ぶかを、活用と探索を両立させながら決める枠組みです。当たりの確率が分からないスロットマシンが並んでいる状況にたとえられます。 $\epsilon$ -greedy方策やUCB方策がここに含まれます。

## 方策勾配法

価値を見積もるのではなく、方策そのものを勾配にしたがって直接改善していく方法です。行動の選択肢が連続した値を取る場合、たとえばロボットの関節の角度のような場合に向いています。REINFORCEが代表です。

## マルコフ決定過程

次にどの状態へ移るかが、今の状態と今取る行動だけで決まるとみなす枠組みです。それより前の経緯は考えません。この前提の上に強化学習の理論が組み立てられているため、必ず押さえておく必要があります。

## 割引率

将来もらえる報酬を、どれくらい低く見積もるかを決める係数です。0に近いほど目先の報酬を重視し、1に近いほど遠い将来まで見ます。無限に続く場合でも合計が発散しないようにする役割もあります。

## SARSA

実際に次にとった行動を使って価値を更新する手法です。自分が現に従っている方策に沿って更新するため、方策オン型と呼ばれます。最大の価値を前提に更新する方策オフ型のQ学習と、対にして違いを押さえます。

# 10 モデルの選択・評価

⇒ **この節の目標。**基本的なモデルの選択基準、評価方法並びに評価指標／訓練誤差と汎化誤差の違い／データの量や目的に応じて、汎化性能を推定する検証方法を適切に選択できる／汎化性能の悪化につながる代表的な現象を列挙・／モデルの適用環境、ビジネス課題、並びに目的に応じて、適切な評価指標・モデルを選択できる／モデルの複雑さを上げることによるメリット・デメリットを理解した上で、モデル選択の指針。

## k-分割交差検証

データをk個のかたまりに分け、そのうち一つを検証用、残りを学習用として評価し、これをk回繰り返して平均を取る方法です。すべてのデータが一度は検証に使われるため、分け方の偏りによる評価のぶれを抑えられます。

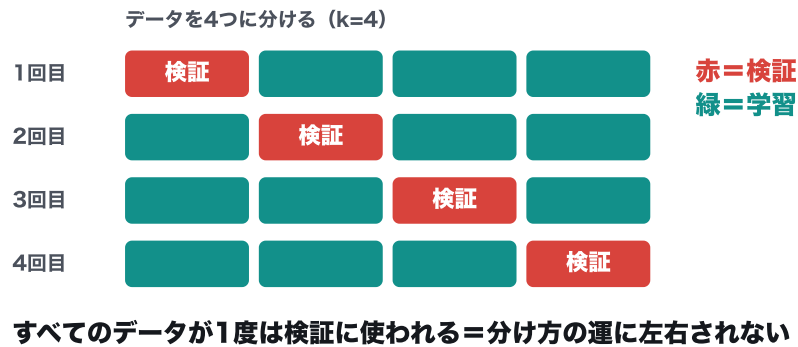


図 k-分割交差検証

### 平均二乗誤差 MSE

予測と実際の差を二乗して平均した値で、回帰の評価に使われます。二乗するため、大きく外した予測を強く罰する性質があります。値が小さいほど良い予測です。ただし単位がもとのデータの二乗になる点には注意が必要です。

### 二乗平均平方根誤差 RMSE

平均二乗誤差の平方根を取った値です。二乗によって変わってしまった単位が、もとのデータと同じ単位に戻るため、どれくらい外しているかを読み取りやすくなります。大きな外れを強く罰する性質はそのまま保たれます。

### 平均絶対値誤差 MAE

予測と実際の差の絶対値を平均した値です。二乗しないため、大きく外れた一部のデータに引きずられにくくなります。外れ値が混ざるデータでは、平均二乗誤差よりこちらの方が実態に合う場合があります。

### ROC曲線・AUC

判定のしきい値を動かしたときに、真陽性率と偽陽性率がどう変わるかを描いた曲線がROC曲線です。その曲線の下側の面積がAUCで、1に近いほど良い性能を示します。データの偏りに強い評価指標として使われます。

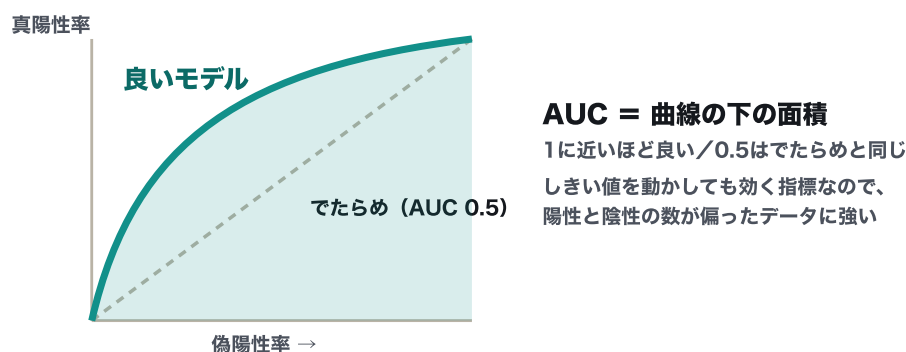


図 ROC曲線・AUC

## 赤池情報量規準 AIC

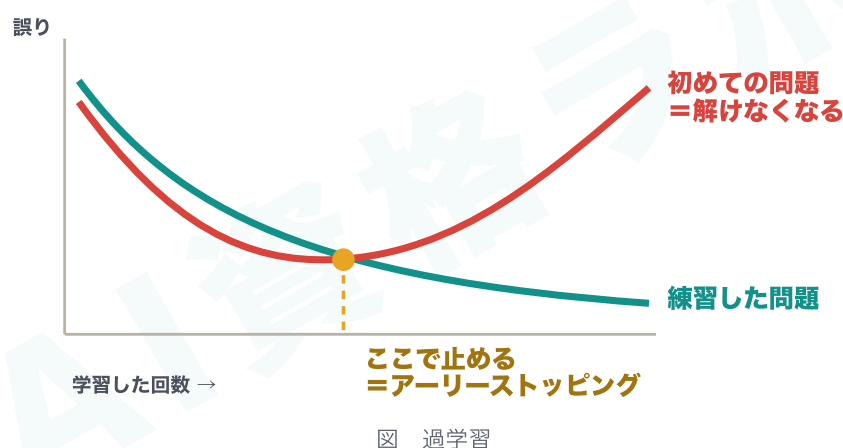
データへの当てはまりの良さと、使ったパラメータの多さの両方を見て、モデルを選ぶための基準です。複雑なモデルには罰を与えるため、当てはまりだけを見て過学習したモデルを選ぶことを防げます。値が小さいほど良いモデルです。

## オッカムの剃刀

同じことを説明できるのであれば、より単純な方を選ぶべきだ、という考え方です。もとは中世の哲学に由来する言葉です。モデル選択では、必要以上に複雑なモデルを避ける指針として引かれ、情報量規準の背景にもなっています。

## 過学習

学習に使ったデータに合わせすぎてしまい、初めて見るデータに対する精度が落ちてしまうことです。学習データでの誤差は下がり続けるのに、検証データでの誤差は途中から上がり始める、という形で現れます。



## 交差検証

データを分けて学習と検証を繰り返し、性能を偏りなく見積もる方法の総称です。一度だけ分けるホールドアウト検証に比べ、分け方の運によるぶれが小さくなります。代表的なやり方がk-分割交差検証です。

## 偽陽性・偽陰性

偽陽性は、陽性と判定したのに実際は陰性だった場合です。偽陰性は、陰性と判定したのに実際は陽性だった場合です。病気の見逃しにあたる偽陰性と、誤って警告する偽陽性は、どちらを重く見るかが場面によって変わります。

## 真陽性・真陰性

真陽性は、陽性と判定して実際も陽性だった場合です。真陰性は、陰性と判定して実際も陰性だった場合です。どちらも判定と実際が一致した場合であり、混同行列では対角線に並びます。正解率はこの二つの合計から求めます。

## 混同行列

予測と実際の組み合わせを、真陽性・偽陽性・偽陰性・真陰性の四つに整理した表です。正解率、適合率、再現率、F値といった指標は、すべてこの表の数字から計算されます。評価の出発点になる表です。



図 混同行列

## 正解率・適合率・再現率・F値

正解率は全体のうち正しく判定できた割合です。適合率は陽性と判定したうち本当に陽性だった割合、再現率は本当に陽性のうち見つけられた割合を指します。適合率と再現率は片方を上げると片方が下がるため、その調和平均であるF値で釣り合いを見ます。

## 汎化性能

学習に使っていない、初めて見るデータに対してどれだけうまく働くかという性能です。機械学習が本当に目指しているのはこの性能であって、学習データでの成績が良いことは目的ではありません。過学習はこの性能が落ちた状態です。

## ベイズ情報量規準 BIC

赤池情報量規準と同じく、当てはまりと複雑さの両方からモデルを選ぶ基準です。データの数が多いほど、複雑さに対する罰が大きくなる点が異なります。そのため、AICより単純なモデルが選ばれやすくなります。

## ホールドアウト検証

データを学習用と検証用に一度だけ分けて評価する、最も単純な方法です。手順が軽く済む一方で、たまたまの分け方によって評価が上下するという弱点があります。データが少ないときほど、その影響が大きくなります。

AI資格ラボ

## 確認問題 第3部のふりかえり

### 確認問題 第3部

#### 第1問 ロジスティック回帰についての記述として、最も適切な選択肢を1つ選べ

答

**名前は回帰だが分類に使い、シグモイド関数で確率を出す**

ロジスティック回帰は線形回帰の出力をシグモイド関数に通し、0から1の値を確率とみなして分けます。結果を確率として解釈できるため、医療や与信でよく使われます。

### 確認問題 第3部

#### 第2問 自己回帰モデルとベクトル自己回帰モデルの違いとして、最も適切な選択肢を1つ選べ

答

**ベクトル自己回帰モデルは、複数の時系列が互いに影響し合う形に広げたものである**

自己回帰モデルは過去の自分の値から次を予測します。ベクトル自己回帰モデルは、金利と株価のように複数の系列が互いに影響し合う関係を同時に扱えるよう広げたものです。

### 確認問題 第3部

#### 第3問 バギングとブースティングの違いとして、最も適切な選択肢を1つ選べ

答

**バギングは並列に学習でき、ブースティングは順番に学習する**

バギングは重複を許して選び直したデータで複数のモデルを独立に作るため並列化できます。ブースティングは前のモデルの誤りを重く見て順に作るため、並列化できません。

## 62語チェックリスト

自分の言葉で説明できたら✓を入れてください。読めるだけでは足りません。

- |                                           |                                              |                                           |
|-------------------------------------------|----------------------------------------------|-------------------------------------------|
| <input type="checkbox"/> アンサンブル学習         | <input type="checkbox"/> 多クラス分類              | <input type="checkbox"/> ウォード法            |
| <input type="checkbox"/> カーネル             | <input type="checkbox"/> バギング                | <input type="checkbox"/> 協調フィルタリング        |
| <input type="checkbox"/> カーネルトリック         | <input type="checkbox"/> ブースティング             | <input type="checkbox"/> クラスタリング          |
| <input type="checkbox"/> 回帰問題             | <input type="checkbox"/> ブートストラップサンプリング      | <input type="checkbox"/> コールドスタート問題       |
| <input type="checkbox"/> 決定木              | <input type="checkbox"/> 分類問題                | <input type="checkbox"/> コンテンツベースフィルタリング  |
| <input type="checkbox"/> 勾配ブースティング        | <input type="checkbox"/> ベクトル自己回帰モデル(VARモデル) | <input type="checkbox"/> 次元削減             |
| <input type="checkbox"/> サポートベクターマシン(SVM) | <input type="checkbox"/> マージン最大化             | <input type="checkbox"/> 主成分分析(PCA)       |
| <input type="checkbox"/> 線形回帰             | <input type="checkbox"/> ランダムフォレスト           | <input type="checkbox"/> 潜在的ディリクレ配分法(LDA) |
| <input type="checkbox"/> 自己回帰モデル(ARモデル)   | <input type="checkbox"/> ロジスティック回帰           | <input type="checkbox"/> 多次元尺度構成法(MDS)    |
| <input type="checkbox"/> 単回帰分析            | <input type="checkbox"/> k-means法            | <input type="checkbox"/> デンドログラム(樹形図)     |
| <input type="checkbox"/> 重回帰分析            | <input type="checkbox"/> t-SNE               | <input type="checkbox"/> 特異値分解(SVD)       |

- |                                               |                                          |                                         |
|-----------------------------------------------|------------------------------------------|-----------------------------------------|
| <input type="checkbox"/> トピックモデル              | <input type="checkbox"/> マルコフ決定過程        | <input type="checkbox"/> 過学習            |
| <input type="checkbox"/> Actor-Critic         | <input type="checkbox"/> 割引率             | <input type="checkbox"/> 交差検証           |
| <input type="checkbox"/> $\epsilon$ -greedy方策 | <input type="checkbox"/> SARSA           | <input type="checkbox"/> 偽陽性・偽陰性        |
| <input type="checkbox"/> REINFORCE            | <input type="checkbox"/> k-分割交差検証        | <input type="checkbox"/> 真陽性・真陰性        |
| <input type="checkbox"/> Q学習                  | <input type="checkbox"/> 平均二乗誤差(MSE)     | <input type="checkbox"/> 混同行列           |
| <input type="checkbox"/> UCB方策                | <input type="checkbox"/> 二乗平均平方根誤差(RMSE) | <input type="checkbox"/> 正解率・適合率・再現率・F値 |
| <input type="checkbox"/> 行動価値関数               | <input type="checkbox"/> 平均絶対値誤差(MAE)    | <input type="checkbox"/> 汎化性能           |
| <input type="checkbox"/> 状態価値関数               | <input type="checkbox"/> ROC曲線・AUC       | <input type="checkbox"/> ベイズ情報量規準(BIC)  |
| <input type="checkbox"/> バンディットアルゴリズム         | <input type="checkbox"/> 赤池情報量規準(AIC)    | <input type="checkbox"/> ホールドアウト検証      |
| <input type="checkbox"/> 方策勾配法                | <input type="checkbox"/> オッカムの剃刀         |                                         |

AI資格ラボ

## 第4部

# ディープラーニングの概要

技術分野／6の中項目・52語

- 11 ニューラルネットワークとディープラーニング (6語)
- 12 活性化関数 (6語)
- 13 誤差関数 (5語)
- 14 正則化 (7語)
- 15 誤差逆伝播法 (4語)
- 16 最適化手法 (24語)

## 11 ニューラルネットワークとディープラーニング

⇒ **この節の目標。**ニューラルネットワークの基礎的な知識／ニューラルネットワークとディープラーニングの関係／ディープラーニングの学習に必要なデータ量や計算リソースについて理解し、ディープラーニングが適用可能な場面を挙げることができる／CPU・GPU・TPUの特徴をそれぞれ／GPUやTPUがディープラーニングの学習・推論に適する理由。

### CPU

コンピュータの中心となる汎用の計算装置です。条件分岐の多い複雑な処理を、順番に手早くこなすのが得意です。一方で、同じ計算を大量に並べて行うのは苦手で、ディープラーニングの学習には向きません。

### GPU

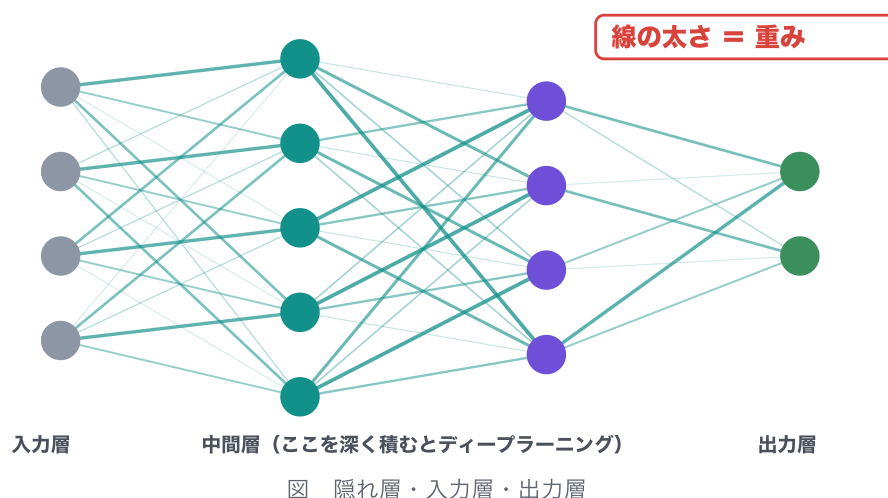
もとは画像処理のために作られた演算装置です。単純な計算を大量に並列で実行できるため、行列の掛け算の塊であるディープラーニングの学習に非常に向いています。NVIDIAが提供するCUDAという開発環境が、普及を大きく後押ししました。

### TPU

Googleがディープラーニング専用開発した演算装置です。行列演算に用途を絞り込むことで、GPUよりもさらに効率よく計算できます。汎用性を捨てて特定の計算に特化させた、専用ハードウェアの代表例です。

### 隠れ層・入力層・出力層

ニューラルネットワークを構成する三種類の層です。入力層がデータを受け取り、出力層が結果を出し、その間にあるのが隠れ層、別名で中間層です。この隠れ層を深く重ねたものが、ディープラーニングと呼ばれます。



## 多層パーセプトロン

単純パーセプトロンに隠れ層を加えたものです。層を重ねることで直線では分けられない問題、たとえば排他的論理和であるXORも解けるようになりました。単純パーセプトロンの限界を越える答えとして重要です。

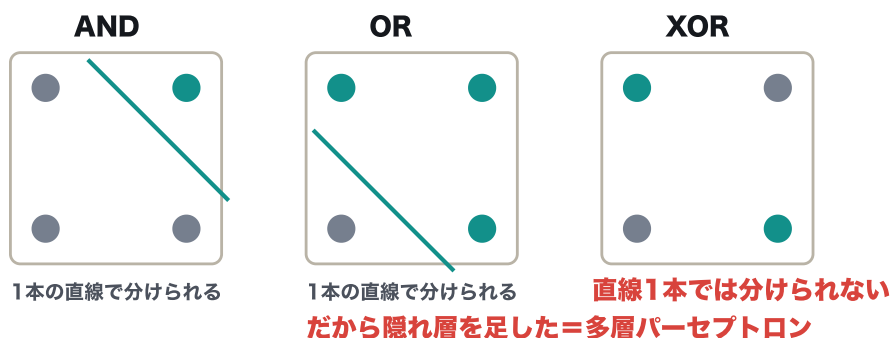


図 多層パーセプトロン

## 単純パーセプトロン

1958年にフランク・ローゼンブラットが考案した、最も基本的なモデルです。入力に重みを掛けて足し合わせ、しきい値を超えたかどうかで出力を決めます。直線ではしか分けられずXORが解けないとミンスキーらに指摘され、第一次ブームの終わりにつながりました。

## 12 活性化関数

⇒ **この節の目標。** 代表的な活性化関数の定義・使い分け・注意点について、それぞれ／ディープラーニングにおける活性化関数の役割。

### Leaky ReLU関数

ReLU関数の、入力が負の側をわずかな傾きに変えたものです。ReLUでは負の入力に対して出力も勾配も0になり、そのノードが学習されなくなることがあります。わずかでも傾きを残すことで、その問題を和らげます。

### ReLU関数

入力が0以下なら0を、正ならその値をそのまま返す活性化関数です。正の側で勾配が常に1のままなので、層を重ねても勾配が小さくなりにくく、勾配消失問題を大きく改善しました。現在の標準的な活性化関数です。

### tanh関数

マイナス1から1までの値を返す、S字型の活性化関数です。0から1を返すシグモイド関数に比べて勾配が大きくなるため、勾配消失が起こりにくくなります。それでも両端では勾配が小さくなる点は変わりません。

## シグモイド関数

0から1までの値を返す、S字型の活性化関数です。出力を確率として扱えるため、二値分類の出力層で使われます。ただし入力が大きくても小さくても勾配がほぼ0になるため、層を重ねると勾配消失問題を引き起こします。

## ソフトマックス関数

複数の出力を、合計がちょうど1になるように変換する関数です。それぞれの値を、そのクラスである確率として読めるようになります。三つ以上に分ける多クラス分類の出力層で使われ、誤差関数には交差エントロピーを組み合わせるのが定石です。

## 勾配消失問題

層を深く重ねると、逆向きに伝えていく勾配がだんだん小さくなり、入力に近い層がほとんど学習されなくなる問題です。両端で勾配がほぼ0になるシグモイド関数が主な原因で、ReLU関数の登場によって大きく改善しました。

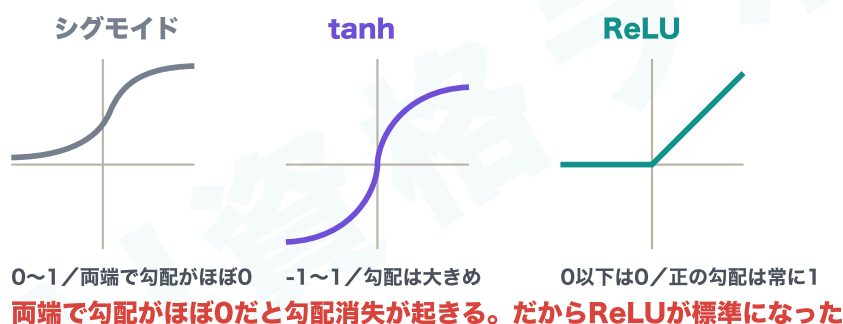


図 勾配消失問題

## 13 誤差関数

⇒ **この節の目標。** 誤差関数の基礎的な知識／代表的な誤差関数／適用するタスクに応じて、適切な誤差関数を選択できる。

### Contrastive Loss

二つのデータが同じ仲間かどうかを学ばせるための誤差関数です。同じ仲間なら距離が近くなるように、違う仲間なら一定以上離れるように働きます。データそのものではなく、距離の取り方を学ぶ距離学習で使われます。

### Triplet Loss

基準となるデータ、同じ仲間のデータ、違う仲間のデータの三つを一組にして使う誤差関数です。基準と仲間の距離が、基準と他人の距離より小さくなるように学習します。顔認証などで広く使われています。

## カルバック・ライブラー情報量 KL

二つの確率分布がどれくらい違うかを測る量です。値は必ず0以上で、二つが完全に同じときだけ0になります。ただし入れ替えると値が変わるため、距離そのものではありません。VAEなどの学習で使われます。

## 交差エントロピー

分類で使われる代表的な誤差関数です。正解のクラスに対してモデルが与えた確率が低いほど、大きな誤差になります。ソフトマックス関数と組み合わせて使われることが多く、分類の標準的な組み合わせです。

## 平均二乗誤差関数

回帰で使われる代表的な誤差関数です。予測と正解の差を二乗して平均します。差を二乗するため、大きく外した予測ほど強く罰せられ、その分だけ大きく修正が入ります。分類で使う交差エントロピーと対にして押さえます。

# 14 正則化

⇒ **この節の目標。** 正則化を導入する目的／代表的な正則化手法の特徴／獲得したいモデルの特性に応じて、適切な正則化手法を選択できる。

## L0正則化

0ではないパラメータの個数そのものを罰する正則化です。不要なパラメータを本当に消せますが、個数を数える計算は微分できず、最適化が非常に難しくなります。そのため実際にはあまり使われません。

## L1正則化

パラメータの絶対値の合計を罰として加える正則化です。不要なパラメータの値がちょうど0になりやすいため、使う特徴を自動的に絞り込む効果があります。これを線形回帰に用いたものがラッソ回帰です。

## L2正則化

パラメータの二乗の合計を罰として加える正則化です。値をちょうど0にはしにくい一方で、全体をなだらかに小さく抑える働きがあります。これを線形回帰に用いたものがリッジ回帰です。重みが極端に大きくなるのを防ぎます。

**L1正則化（ラッソ回帰）**

絶対値の和を罰する

**0になるものが出る＝特徴を選べる****L2正則化（リッジ回帰）**

二乗の和を罰する

**全体が小さくなる＝0にはならない**

棒＝パラメータの大きさ。罰を加えることで、複雑になりすぎるのを防ぐ

図 L2正則化

**正則化**

モデルが複雑になりすぎるのを防ぐために、パラメータの大きさなどに罰を加える工夫です。学習データに合わせすぎる過学習を抑え、初めて見るデータへの性能を上げることが狙いです。L1とL2が代表的です。

**ドロップアウト**

学習のたびに、一部のノードをランダムに使わないようにする手法です。毎回少しずつ違う形のネットワークで学ぶことになり、複数のモデルを組み合わせたのと似た効果が生まれます。過学習を抑える方法として広く使われます。

**ラッソ回帰**

L1正則化を組み込んだ線形回帰です。効いていない説明変数の係数がちょうど0になりやすいため、変数の選択を同時に行えるのが特徴です。説明変数が非常に多く、そのうち一部だけが効いている場合に向いています。

**リッジ回帰**

L2正則化を組み込んだ線形回帰です。係数全体を小さく抑えることで、推定が不安定になるのを防ぎます。説明変数どうしの相関が強く、多重共線性が起きている場合に有効です。ラッソ回帰と違い、係数はちょうど0にはなりません。

**15 誤差逆伝播法**

⇒ **この節の目標。** 誤差逆伝播法の概要／誤差逆伝播法の適用時に生じる問題とその主たる原因について。

**勾配消失問題**

誤差逆伝播法の側から見ると、連鎖律にしたがって各段の微分を掛け合わせていくために起こる問題です。1より小さい値が何度も掛かると、入力に近い層へ届くころには勾配がほぼ0になり、そこが学習されなくなります。

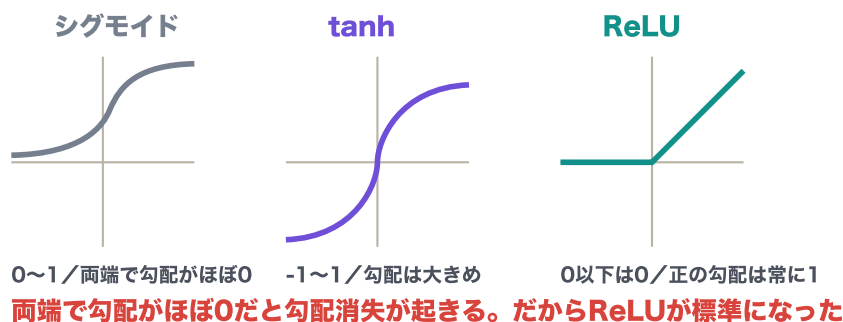


図 勾配消失問題

### 勾配爆発問題

勾配消失とは逆に、1より大きい値が何度も掛かって、勾配が発散してしまう問題です。同じ重みを繰り返し使う回帰結合層で起こりやすく、勾配の大きさに上限を設ける勾配クリッピングで抑えます。

### 信用割当問題

最後に出た結果の良し悪しに対して、どの層のどの重みがどれだけ貢献したのかを割り振る、という問題です。層が深いほど難しくなります。誤差逆伝播法は、この問題を数学的に解く方法として位置づけられます。

### 連鎖律

合成関数の微分を、各段階の微分の掛け算として求める規則です。ニューラルネットワークは関数を何段も重ねたものなので、この規則によって出力側から入力側へ順に微分を求められます。誤差逆伝播法の数学的な土台です。

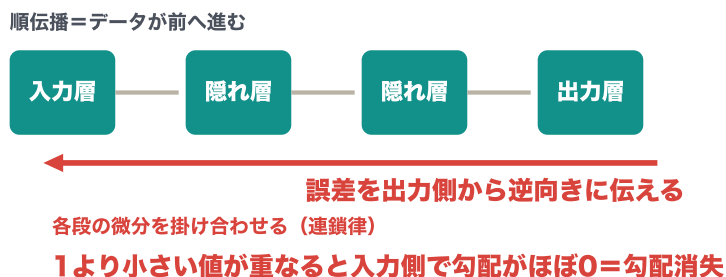


図 連鎖律

## 16 最適化手法

⇒ **この節の目標。** 勾配降下法の概要／勾配降下法の問題とそれを解決するための手法を列挙できる／勾配降下法の計算を効率化する方法／ハイパーパラメータの概要と代表的な調整方法を列挙・。

## AdaBound

Adamの学習率に上限と下限を設ける手法です。学習の初めはAdamのように速く進み、終わりに近づくにつれて確率的勾配降下法のように振る舞います。Adamが終盤で安定しにくい弱点を補うことを狙っています。

## AdaDelta

AdaGradの、学習が進むほど学習率が下がりすぎるという弱点を改善した手法です。過去すべてではなく、直近の勾配だけを見るようにすることで、学習率が0に近づいて止まってしまうのを防ぎます。

## AdaGrad

パラメータごとに学習率を変える手法です。これまでよく更新されてきたパラメータほど学習率を下げ、あまり動いていないものは大きく動かします。ただし過去の勾配を足し続けるため、進むほど学習率が小さくなりすぎます。

## Adam

モーメンタムと、RMSpropによる学習率の調整を組み合わせた手法です。設定をあまり調整しなくてもよく働くため、現在最も広く使われています。まず試す既定の選択肢として押さえておきます。

## AMSBound

AMSGradに、AdaBoundと同じように学習率の上下の制限を加えた手法です。学習の初めは適応的に速く進み、終わりに近づくと確率的勾配降下法のように安定して振る舞います。AdaBoundと対にして覚えます。

## RMSprop

AdaGradの学習率が下がりすぎる弱点を、過去の勾配を指数的に忘れていくことで改善した手法です。直近の勾配を重く見るため、学習率が小さくなりすぎずに済みます。Adamの構成要素の一つでもあります。

## 鞍点

ある方向から見ると谷底なのに、別の方向から見ると山の頂上になっている点です。馬の鞍の形に似ていることからこう呼ばれます。勾配が0になるため学習が止まりやすく、次元が高いほど局所最適解よりこちらが問題になります。

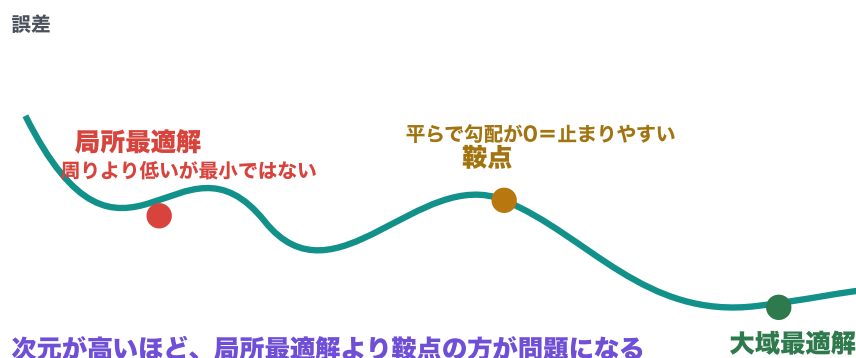


図 鞍点

## イテレーション

重みを一回更新することを指します。ミニバッチ学習では、一つのミニバッチを処理して更新するたびにイテレーションです。学習データ全体を一巡するエポックと、混同しやすいので区別して覚えましょう。

## エポック

学習データ全体を一通り学習し終えることを指します。エポック数を増やすほど学習は進みますが、増やしすぎると学習データに合わせすぎて過学習に陥ります。重みを一回更新するイテレーションとは意味が違うので、区別が必要です。

## オンライン学習

データを一つずつ使って、その都度重みを更新していく方法です。データが届くたびに学び続けられるため、状況が変わり続ける場面に向きます。一方で、一つのデータに引きずられて更新が不安定になりやすい性質があります。

## 学習率

一回の更新でどれだけ大きくパラメータを動かすかを決める値です。大きすぎると最小値を乗り越えて発散し、小さすぎるといつまでも学習が進みません。人が事前に決めるハイパーパラメータの代表格です。

## 確率的勾配降下法 SGD

学習データ全体ではなく、ランダムに選んだ一つ、あるいは少数のデータだけを使って更新する方法です。一回の計算が軽く、更新にばらつきがあるぶん、局所最適解から抜け出しやすいという利点もあります。

## グリッドサーチ

ハイパーパラメータの候補を格子状に並べ、その組み合わせをすべて試す方法です。網羅的に探せる代わりに、パラメータの数が増えると組み合わせが爆発的に増え、現実的な時間で終わらなくなります。ランダムサーチと対比されます。

## 勾配降下法

誤差が小さくなる方向、つまり勾配の逆向きへ、少しずつパラメータを動かしていく方法です。ニューラルネットワークの学習の土台になる考え方で、一回にどれだけ動かすかを決めるのが学習率です。山を下りることにたとえられます。

## 局所最適解

その周りと比べれば最も低いものの、全体で見れば最小ではない点です。勾配降下法はここに落ち込むと抜け出せなくなります。更新にばらつきを持たせる確率的勾配降下法は、この対策としても働きます。

## 早期終了

検証用データに対する誤差が下がり止まって上がり始めたところで、学習を打ち切ることです。学習データへの当てはめが進みすぎる前に止めるため、過学習を防ぐ簡単で有効な方法として使われます。

## 大域最適解

全体を通して本当に最小となる点です。学習が目指す理想の到達点ですが、複雑なモデルでは、そこに到達したかどうかを確かめる手立てがありません。周りだけを見れば最小である局所最適解と、必ず区別して覚えます。

## 二重降下現象

モデルを大きくしていくと、性能がいったん悪くなった後、さらに大きくすると再び良くなるという現象です。複雑にしすぎると必ず悪化するという従来の常識と食い違うため、近年注目されています。

## ノーフリーランチの定理

あらゆる問題に対して万能に優れた手法は存在しない、という定理です。ある問題で優れた手法は、別の問題では劣ります。だからこそ、問題に応じて手法を選び、試して比べる必要がある、という根拠になります。

## ハイパーパラメータ

学習によって決まるのではなく、人があらかじめ決めておく設定値です。学習率、層の数、ミニバッチの大きさなどが当たります。良い値を見つけるために、グリッドサーチやランダムサーチが使われます。

## バッチ学習

学習データ全体を使って誤差を計算し、一回だけ重みを更新する方法です。更新の向きが安定する一方、データが大きいと一回の計算が重く、メモリにも収まりません。一括学習とも呼ばれ、ミニバッチ学習と対比されます。

## ミニバッチ学習

学習データを小さなまとまりに分け、そのまとまりごとに更新する方法です。バッチ学習の安定さと、オンライン学習の軽さの中間に当たります。現在の標準的なやり方で、まとまりの大きさもハイパーパラメータです。

## モーメンタム

前回の更新の向きを、慣性として今回の更新に加える工夫です。谷を左右に往復するような振動が抑えられ、同じ向きに進み続ける場合には加速します。物理の慣性にたとえた考え方で、Adamにも組み込まれています。

## ランダムサーチ

ハイパーパラメータの組み合わせをランダムに選んで試す方法です。格子状に全部試すグリッドサーチより、同じ試行回数でも良い値に当たりやすい場合があることが知られています。効くパラメータが一部に限られるときに有利です。

## 確認問題 第4部のふりかえり

### 確認問題 第4部

**第1問** 以下の文章を読み、空欄に最もよく当てはまる選択肢を1つ選べ Googleがディープラーニング専用開発し、行列演算に用途を絞ることでGPUよりさらに効率よく計算できるようにした演算装置を（ ）という

答

**TPU**

TPUはGoogleがディープラーニング専用開発した演算装置です。汎用性を捨てて特定の計算に特化させた、専用ハードウェアの代表例です。

### 確認問題 第4部

**第2問** 単純パーセプトロンの限界と、その解決について、最も適切な選択肢を1つ選べ

答

**直線ではしか分けられずXORが解けなかったが、隠れ層を加えることで解けるようになった**

単純パーセプトロンは1958年にローゼンブラットが考案しましたが、直線ではしか分けられずXORが解けないとミンスキーらに指摘されました。隠れ層を加えた多層パーセプトロンで解けるようになりました。

### 確認問題 第4部

**第3問** 勾配消失問題が起こりやすくなる条件として、最も適切な選択肢を1つ選べ

答

**ネットワークの層が深く、両端で勾配がほぼ0になる活性化関数を使っている**

層を重ねると逆向きに伝える勾配が掛け合わされます。シグモイド関数のように両端で勾配がほぼ0になる関数では、入力に近い層まで届くころにほぼ0になります。ReLU関数の登場で大きく改善しました。

## 52語チェックリスト

自分の言葉で説明できたら✓を入れてください。読めるだけでは足りません。

- |                                       |                                           |                                              |
|---------------------------------------|-------------------------------------------|----------------------------------------------|
| <input type="checkbox"/> CPU          | <input type="checkbox"/> ReLU関数           | <input type="checkbox"/> カルバック・ライブラー情報量 (KL) |
| <input type="checkbox"/> GPU          | <input type="checkbox"/> tanh関数           | <input type="checkbox"/> 交差エントロピー            |
| <input type="checkbox"/> TPU          | <input type="checkbox"/> シグモイド関数          | <input type="checkbox"/> 平均二乗誤差関数            |
| <input type="checkbox"/> 隠れ層・入力層・出力層  | <input type="checkbox"/> ソフトマックス関数        | <input type="checkbox"/> L0正則化               |
| <input type="checkbox"/> 多層パーセプトロン    | <input type="checkbox"/> 勾配消失問題           | <input type="checkbox"/> L1正則化               |
| <input type="checkbox"/> 単純パーセプトロン    | <input type="checkbox"/> Contrastive Loss | <input type="checkbox"/> L2正則化               |
| <input type="checkbox"/> Leaky ReLU関数 | <input type="checkbox"/> Triplet Loss     | <input type="checkbox"/> 正則化                 |

- |                                   |                                        |                                      |
|-----------------------------------|----------------------------------------|--------------------------------------|
| <input type="checkbox"/> ドロップアウト  | <input type="checkbox"/> AMSBound      | <input type="checkbox"/> 早期終了        |
| <input type="checkbox"/> ラッソ回帰    | <input type="checkbox"/> RMSprop       | <input type="checkbox"/> 大域最適解       |
| <input type="checkbox"/> リッジ回帰    | <input type="checkbox"/> 鞍点            | <input type="checkbox"/> 二重降下現象      |
| <input type="checkbox"/> 勾配消失問題   | <input type="checkbox"/> イテレーション       | <input type="checkbox"/> ノーフリーランチの定理 |
| <input type="checkbox"/> 勾配爆発問題   | <input type="checkbox"/> エポック          | <input type="checkbox"/> ハイパーパラメータ   |
| <input type="checkbox"/> 信用割当問題   | <input type="checkbox"/> オンライン学習       | <input type="checkbox"/> バッチ学習       |
| <input type="checkbox"/> 連鎖律      | <input type="checkbox"/> 学習率           | <input type="checkbox"/> ミニバッチ学習     |
| <input type="checkbox"/> AdaBound | <input type="checkbox"/> 確率的勾配降下法(SGD) | <input type="checkbox"/> モーメンタム      |
| <input type="checkbox"/> AdaDelta | <input type="checkbox"/> グリッドサーチ       | <input type="checkbox"/> ランダムサーチ     |
| <input type="checkbox"/> AdaGrad  | <input type="checkbox"/> 勾配降下法         |                                      |
| <input type="checkbox"/> Adam     | <input type="checkbox"/> 局所最適解         |                                      |

AI資格ラボ

## 第5部

# ディープラーニングの要素技術

技術分野／9の中項目・60語

- 17 全結合層 (2語)
- 18 畳み込み層 (10語)
- 19 正規化層 (4語)
- 20 プーリング層 (4語)
- 21 スキップ結合 (1語)
- 22 回帰結合層 (12語)
- 23 Attention (10語)
- 24 オートエンコーダ (5語)
- 25 データ拡張 (12語)

## 17 全結合層

→ **この節の目標。** 全結合層の概要／全結合層のパラメータ数について／ディープラーニングにおける全結合層の役割。

### 重み

入力にかける係数のことで、学習によって調整される値そのものです。重みが大きいほど、その入力を重視して判断していることになります。ニューラルネットワークが学習するとは、この重みを少しずつ更新していくことを指します。

### 線形関数

入力に重みを掛けて足し合わせるだけの変換です。全結合層が行っているのはこの計算です。ただし線形関数をいくら重ねても、まとめれば一つの線形関数にしかありません。だから層の間に、非線形な活性化関数を挟む必要があります。

## 18 畳み込み層

→ **この節の目標。** 畳み込み層の基礎的な知識／全結合層と畳み込み層の差異について、／畳み込み層の役割について／畳み込み層のパラメータ数について／畳み込み層が適用できるデータの特性について。

### Atrous Convolution

フィルタの要素の間に隙間を空けて畳み込む方法です。Dilated Convolutionと同じものを指します。パラメータの数を増やさずに、一度に見渡せる範囲を広げられるため、画像全体の文脈をつかみたい場面で使われます。

### Depthwise Separable Convolution

畳み込みを、空間方向の処理とチャネル方向の処理の二段階に分ける方法です。まとめて行う通常の畳み込みに比べ、計算量とパラメータを大幅に減らせます。スマートフォン向けのMobileNetで採用されました。

### Dilated Convolution

Atrous Convolutionと同じで、フィルタに隙間を空けて広い範囲を見る畳み込みです。呼び名が二つある点がそのまま問われることがあります。画像の意味を画素ごとに分けるセグメンテーションでよく使われます。

### カーネル

畳み込みで使う小さな行列のことで、フィルタとも呼ばれます。注意したいのは、サポートベクターマシンで出てきたカーネルとは全くの別物だという点です。あちらは高い次元へ写す関数、こちらは画像の上を滑らせる小さな行列です。

## ストライド

フィルタを一度にどれだけずらすかという幅です。1なら1画素ずつ、2なら2画素ずつ動かします。大きくするほど出力の特徴マップは小さくなり、計算も軽くなりますが、細かい情報は失われます。

## 畳み込み操作

フィルタを画像の上で滑らせながら、重なった位置どうしを掛けて足し合わせる操作です。同じフィルタを画像のどこでも使い回すため、重みの数を大きく減らせます。これが全結合層との根本的な違いです。

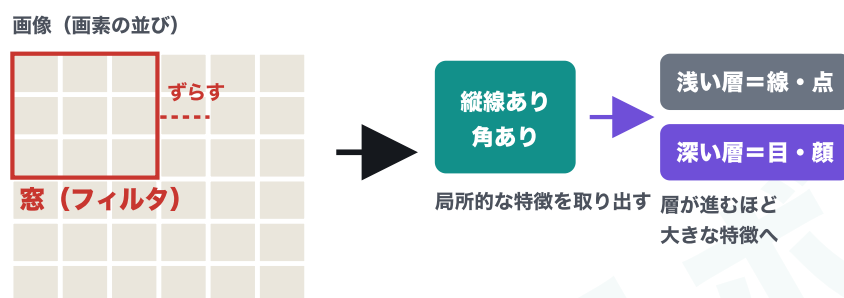


図 畳み込み操作

## 畳み込みニューラルネットワーク CNN

畳み込み層とプーリング層を重ねて作るネットワークです。近くの画素どうしの関係を捉え、位置が多少ずれても同じものと判断できます。画像認識の標準的な構造であり、源流は福島邦彦のネオコグニロンにさかのぼります。

## 特徴マップ

畳み込み操作の出力にあたる、二次元の並びです。フィルタが反応した場所ほど大きな値になるため、その特徴が画像のどこにあるかを表した地図のように読めます。フィルタの数だけ特徴マップができます。

## パディング

入力の周りを0などで埋めてから畳み込むことです。埋めないと端の画素が中心になる機会が少なく、情報が失われます。また出力が入力より小さくなってしまうため、大きさをそろえたいときにも使われます。

## パディングなし

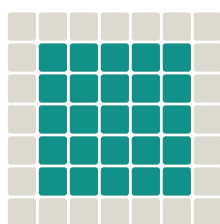


5×5 → 出力3×3

端の画素が中心になりにくい → 出力も5×5のまま

赤=フィルタが当たる位置

## パディングあり



まわりを0で埋める

→ 出力も5×5のまま

## ストライド2



2つずつ飛ばして動かす

出力が小さく、計算も軽い

図 パディング

## フィルタ

畳み込みで画像の上を滑らせる小さな行列で、カーネルと同じものです。どんな特徴に反応するかは人が決めるのではなく、学習によって決まります。浅い層では線や角、深い層ではより複雑な形に反応するようになります。

## 19 正規化層

⇒ **この節の目標。**正規化層の基礎的な知識／代表的な正規化手法について／正規化層がディープラーニングモデルの学習において、どのような役割を果たすのか。

## グループ正規化

チャンネルをいくつかの組に分け、その組ごとに正規化する方法です。バッチ正規化とレイヤー正規化の中間に当たります。バッチの大きさに影響されないため、大きな画像を扱ってバッチを小さくせざるを得ない場面で使われます。

## バッチ正規化

ミニバッチの中で、平均が0、分散が1になるように整える方法です。学習が速く安定し、勾配消失も起きにくくなります。ただしバッチが小さいと統計量が当てにならず、かえって不安定になるという弱点があります。

## レイヤー正規化

一つのデータの中で、すべてのチャンネルをまとめて正規化する方法です。バッチの大きさに全く影響されないため、系列の長さが変わる自然言語処理に向いています。Transformerで使われているのはこの方式です。

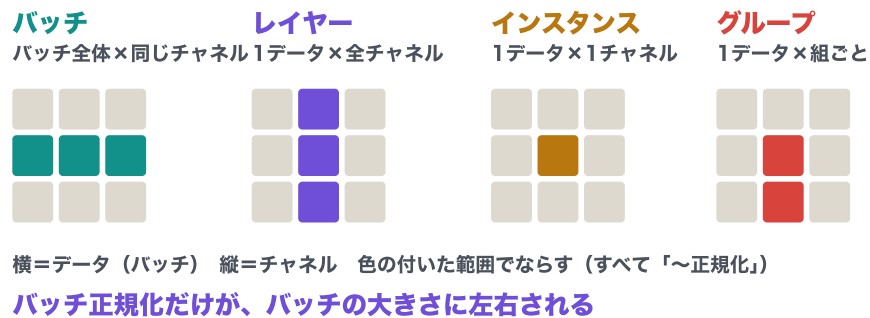


図 レイヤー正規化

## インスタンス正規化

一つのデータの、一つのチャンネルごとに正規化する方法です。画像それぞれが持つ明るさや色合いの偏りを打ち消す働きがあるため、絵の画風を別の画風に変える画風変換のような場面で使われます。バッチの大きさには影響されません。

## 20 プーリング層

⇒ **この節の目標。** プーリング層の基礎的な知識／代表的なプーリング操作について／プーリング層がディープラーニングモデルの学習において、どのような役割を果たすのか。

### グローバルアベレージプーリング GAP

特徴マップ全体の平均を取って、一つの値にまとめる方法です。最後の全結合層の代わりに使うことで、パラメータの数を大きく減らし、過学習を抑えられます。どこに反応したかを可視化する手法の土台にもなっています。

### 最大値プーリング

決められた範囲の中で、最も大きい値だけを取り出す方法です。最も強く反応した場所の情報だけが残るため、特徴がはっきりと保たれます。プーリングの方式としては最もよく使われ、位置のずれに強くする働きを担います。

### 不変性の獲得

対象の位置が少しずれても、出力が変わらない性質を身につけることです。プーリング層が担う役割で、これによって同じ猫が画像の左にいても右にいても同じ猫だと判断できます。畳み込みと組み合わせで効いてくる性質です。



**位置が少しずれても同じ値になる＝不変性の獲得**

猫が画像の左にいても右にいても、同じ猫と判断できるようになる

図 不変性の獲得

## 平均値プーリング

決められた範囲の値の平均を取る方法です。最大値プーリングに比べて、範囲全体の情報がなだらかに残ります。最も強い反応だけを見たいのか、全体の傾向を見たいのかによって、二つを使い分けることになります。

## 21 スキップ結合

⇒ **この節の目標。** スキップ結合の概要／ディープラーニングにおけるスキップ結合の役割／スキップ結合を活用したResNetの概要。

### ResNet

層をいくつか飛び越えて入力をそのまま足す、スキップ結合を導入したネットワークです。これによって勾配が浅い層まで届きやすくなり、100層を超える深さでも学習できるようになりました。2015年のILSVRCで優勝しています。

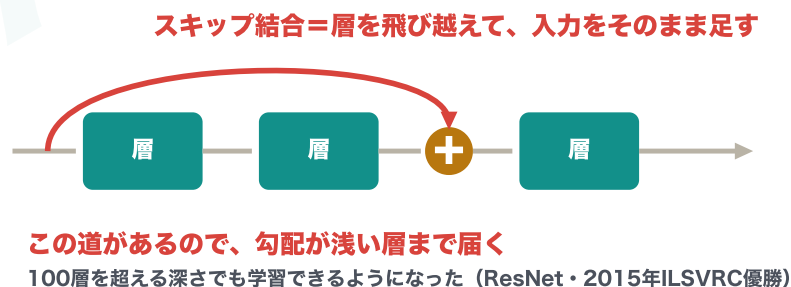


図 ResNet

## 22 回帰結合層

⇒ **この節の目標。** 回帰結合層の概要／回帰結合層を含むネットワークであるリカレントニューラルネットワーク(RNN)／リカレントニューラルネットワーク(RNN)がどのような特性のデータに適したモデルか／リカレントニューラルネットワーク(RNN)の学習方法について／リカレントニューラルネットワーク(RNN)の学習における課題とその解決手法。

## BPTT

時間方向に展開したうえで誤差逆伝播法を行う方法です。回帰結合層は同じ重みを何度も使うため、時間をさかのぼって誤差を伝えます。系列が長いほど掛け算の回数が増え、勾配消失や勾配爆発が起きやすくなります。

## GRU

LSTMを簡単にしたゲート機構です。LSTMが三つのゲートを持つのに対し、GRUは二つにまとめられています。パラメータが少ないぶん学習が速く、性能はLSTMと大きくは変わらない場合が多いとされます。

## LSTM

長い間隔を空けた依存関係を覚えていられるようにした回帰結合層です。情報をためておく記憶セルと、入力ゲート・出力ゲート・忘却ゲートの三つの門を持ちます。これによって勾配消失問題を大きく和らげました。

## エルマンネットワーク

隠れ層の出力を、次の時刻の入力として戻す形の回帰結合ネットワークです。出力層の値を戻すジョルダンネットワークと対にして問われます。現在のRNNは、こちらのエルマン型が基本になっています。

## 勾配消失問題

回帰結合層では、時間をさかのぼって誤差を伝えるため、系列が長いほど掛け算の回数が増えます。1より小さい値が重なると勾配がほぼ0になり、遠い過去の情報が学習に反映されません。この対策として作られたのがLSTMです。

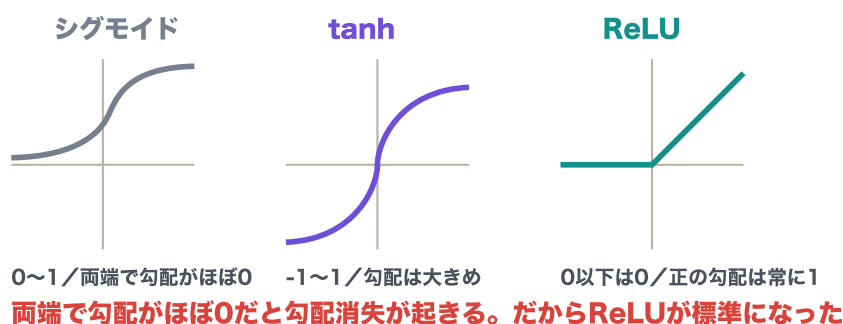


図 勾配消失問題

## 勾配爆発問題

逆に1より大きい値が重なって、勾配が発散してしまう問題です。同じ重みを何度も掛ける回帰結合層で特に起きやすくなります。対策としては、勾配の大きさに上限を設ける勾配クリッピングが使われます。

## 教師強制

学習のときに、前の時刻に自分が出した予測ではなく、正解の値を次の入力として与える方法です。学習が速く安定する一方、実際に使うときは自分の予測を使うため、そのずれが誤りの連鎖を招くという弱点があります。

## ゲート機構

情報をどれだけ通し、どれだけ遮るかを、学習によって制御する仕組みです。LSTMやGRUの中心にあり、覚えておくべき情報と忘れてよい情報を選び分けられるようにします。これが長期の依存を扱える鍵になっています。

### RNN : 長いと前を忘れる



「去年の夏に…沖縄で…食べた」の「去年」が薄れていく

### LSTM : 残すものと捨てるものを決める

覚えておく（長期） + いま使う（短期） = 長・短期記憶

図 ゲート機構

## 双方向RNN Bidirectional RNN

系列を前から処理するものと、後ろから処理するものを組み合わせたネットワークです。ある時点の判断に、後ろの文脈も使えるようになります。文全体があらかじめ分かっている場合には有効ですが、逐次予測には使えません。

## 時系列データ

時間の順に並んだデータのことです。株価、気温、音声、文章などが当たります。順番そのものに意味があるため、並べ替えると情報が壊れてしまう点が、通常のデータとの決定的な違いです。回帰結合層はこれを扱うための仕組みです。

## ジョルダンネットワーク

出力層の出力を、次の時刻の入力として戻す形の回帰結合ネットワークです。隠れ層の出力を戻すエルマンネットワークと対になります。どちらの層の値を戻すのかという違いを、必ず区別して覚えます。

## リカレントニューラルネットワーク RNN

前の時刻の情報を保持しながら、系列を順に処理していくネットワークです。同じ重みを各時刻で使い回すため、長さの違う系列も扱えます。一方で、長い系列では勾配消失が起こりやすいという弱点があります。

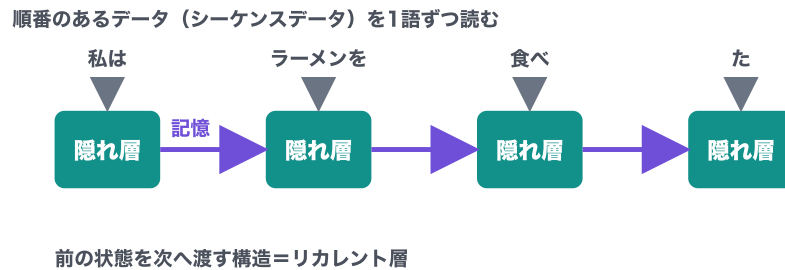


図 リカレントニューラルネットワーク(RNN)

## 23 Attention

⇒ **この節の目標。** Attentionの基礎的な知識／Attentionがどのような役割を果たすのか／リカレントニューラルネットワーク(RNN)の問題点をAttentionはどのように解決できるか／Attentionを用いた代表的モデルのTransformerについて／Self-AttentionとSource-Target Attention(Encoder-Decoder Attention)について。

### Attention

入力のどこに注目すべきかを、重みとして計算する仕組みです。系列を一つの固定長のベクトルに押し込める必要がなくなり、長い文でも情報が失われにくくなりました。現在の言語モデルの土台にあたる考え方です。

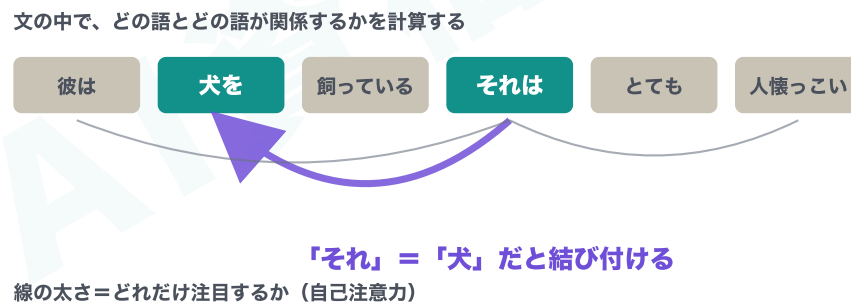


図 Attention

### Multi-Head Attention

Attentionを複数、並列に行う仕組みです。それぞれが違う観点で関係を捉えるため、語の並び方や意味の対応など、複数の関係を同時に扱えます。Transformerの構成要素の一つです。

### Self-Attention

同じ系列の中で、それぞれの要素が他のすべての要素との関係を見る仕組みです。離れた語どうしの関係も一度に捉えられます。順に処理する必要がないため並列に計算でき、Transformerの中心になっています。

## Seq2Seq

入力の系列を一度まとめるエンコーダと、そこから出力の系列を作るデコーダを組み合わせた枠組みです。機械翻訳で使われました。ただし入力を固定長にまとめるため長文に弱く、その弱点を補うために出てきたのがAttentionです。

## Source-Target Attention

デコーダ側が、エンコーダ側の出力に注目する形のAttentionです。翻訳でいえば、出力する語を決めるときに入力文のどこを見るかを決める働きをします。同じ系列の中で見るSelf-Attentionと対になります。

## Transformer

回帰結合を一切使わず、Attentionだけで構成したモデルです。2017年の論文「Attention Is All You Need」で発表されました。順に処理する必要がなくなったため並列化でき、大規模な学習が可能になりました。

### RNN : 1語ずつ順番に



### Transformer : 全部を一度に見る



2017年。いま名前を聞く生成AIは、ほぼ全部この子孫

図 Transformer

## 位置エンコーディング

語の並び順を表す情報を、入力に足し込む仕組みです。Transformerは系列を並列に処理するため、そのままでは語の順序が分かりません。順序という、言語にとって欠かせない情報を補うための工夫です。

### 一度に見ると、順番の情報が消える



### 位置の情報を数値で足す



図 位置エンコーディング

## キー

Attentionで、照合される側が持つ情報です。クエリと突き合わせて、どれだけ関係が深いかを計算するために使われます。クエリ・キー・バリューの三つ組で、辞書を引く動作にたとえられます。

## クエリ

Attentionで、探しに行く側が持つ情報です。これを各キーと突き合わせて、どれだけ関係が深いかを重みとして求めます。辞書を引くときの、調べたい見出し語にあたります。キー、バリューと三つ組で覚えます。

## バリュー

Attentionで、実際に取り出される値です。クエリとキーの照合で求めた重みを使って、バリューを重み付きで足し合わせたものが出力になります。辞書を引いて得られる、語釈にあたります。

## 24 オートエンコーダ

→ **この節の目標。** オートエンコーダの概要／ディープラーニングにおけるオートエンコーダの役割／オートエンコーダの代表的な亜種。

### VQ-VAE・info VAE・ $\beta$ -VAE

変分オートエンコーダを発展させたモデルです。VQ-VAEは潜在変数を連続ではなく離散の値として扱い、 $\beta$ -VAEは正則化の強さを調節して、潜在変数の各要素が別々の意味を担うように促します。

## 次元削減

オートエンコーダは、入力をいったん小さな中間層に押し込めてから元に戻すよう学習します。そのため中間層は、入力情報を保ったまま次元を減らしたものになります。主成分分析を非線形に広げたものの、と説明されることもあります。

## 事前学習

層ごとにオートエンコーダとして学習させ、その重みを本番の学習の初期値として使う手法です。深い層をうまく学習できなかった時代に、その突破口となりました。ReLUなどの登場により、現在ではあまり使われません。

## 積層オートエンコーダ

オートエンコーダを何段も積み重ねたものです。一段ずつ順に事前学習してから、全体をまとめて調整します。ディープラーニングが深い層を学習できるようになった初期の重要な手法として位置づけられます。

## 変分オートエンコーダ VAE

潜在変数を一つの値ではなく、確率分布として学習するオートエンコーダです。分布から値を取り出して復元するため、学習していない新しいデータを作り出せます。生成モデルの代表の一つに数えられます。

入口で縮めて、出口で膨らませる



潜在ベクトル=特徴を圧縮した数値の並び

図 変分オートエンコーダ(VAE)

## 25 データ拡張

⇒ **この節の目標。** データ拡張の基礎的な知識／代表的なデータ拡張手法について／ディープラーニングにおけるデータ拡張の役割／タスクやデータセットに応じて、使用するデータ拡張手法を選択できる。

### Contrast

画像の明暗の差を強めたり弱めたりして、学習データを水増しする方法です。撮影したときの条件の違いに左右されにくいモデルを作れます。データ拡張の中でも、色や明るさに関する基本的な操作の一つに数えられます。

### Brightness

画像全体の明るさを変えて、学習データを水増しする方法です。暗い場所で撮られた写真でも同じものと判断できるようになります。明るさを変えても写っているものは変わらない、という前提の上に成り立ちます。

### Crop

画像の一部を切り抜いて、学習データを水増しする方法です。対象が画面のどこにあっても、また一部しか写ってなくても認識できるようになります。ここで注意したいのは、切り抜きすぎて対象が消えないよう、範囲には注意が必要です。

### CutMix

二枚の画像の一部を切り貼りして混ぜ、正解ラベルも面積の比率で混ぜる方法です。Cutoutのように情報を捨てるのではなく、別の画像で埋める点が特徴です。Mixupと並ぶ、強力なデータ拡張として知られます。

## Cutout

画像の一部を四角く塗りつぶして隠す方法です。一部が隠れていても判断できるようになり、画像の特定の場所だけに頼った学習を防げます。Random Erasingとよく似た考え方で、対にして押さえておきます。

## Mixup

二枚の画像を重み付きで重ね合わせ、正解ラベルも同じ比率で混ぜる方法です。中間の入力に対して中間の答えを返すようになり、判断の境界がなだらかになります。CutMixと対にして押さえます。

## noising

データにわざと雑音を加えて水増しする方法です。多少の乱れがあっても正しく判断できるようになります。画像だけでなく音声やテキストにも使え、雑音を加えた入力から元を復元させるオートエンコーダの学習にも応用されます。

## paraphrasing

文章を、意味を変えずに別の言い方に書き換えて水増しする方法です。自然言語処理におけるデータ拡張にあたります。注意が必要なのは、言い換えによって意味が変わってしまうと、誤った正解を教えることになるため注意が必要です。

## RandAugument

あらかじめ用意した複数の変換の中から、ランダムに選んで適用する方法です。どの変換をどれだけ強くかけるかを探す手間を大きく減らせます。データ拡張の設定探しを自動化する流れの一つです。

## Random Erasing

画像の中の、ランダムな位置と大きさの領域を消してしまう方法です。対象の一部が隠れた状況を人工的に作り出せます。Cutoutと考え方はほぼ同じで、隠す領域の大きさや位置の決め方に違いがあります。

## Random Flip

画像を左右や上下に反転させて水増しする方法です。ただし常に使えるわけではありません。文字や道路標識、数字などは反転すると意味が変わってしまうため、対象によって使えるかどうかを判断する必要があります。

## Rotate

画像を回転させて水増しする方法です。傾いて写っていても認識できるようになります。ただしこれも対象次第で、たとえば数字の6と9のように、回転させると別の意味になってしまうものには使えません。



**やること＝水増し／総称＝データ拡張技術**

中身は変えずに、見え方だけ変える。反転させた猫は、やっぱり猫

☒ Rotate

## 確認問題 第5部のふりかえり

### 確認問題 第5部

#### 第1問 全結合層と活性化関数の関係として、最も適切な選択肢を1つ選べ

答

**線形関数をいくら重ねても一つの線形関数にしかならないため、間に非線形な活性化関数が必要**

全結合層が行うのは、入力に重みを掛けて足す線形な計算です。これをいくら重ねてもまとめれば一つの線形関数にしかならないため、層の間に非線形な活性化関数を挟む必要があります。

### 確認問題 第5部

#### 第2問 畳み込みにおけるカーネルと特徴マップについて、最も適切な選択肢を1つ選べ

答

**カーネルはフィルタと同じ小さな行列で、その出力が特徴マップである**

畳み込みのカーネルはフィルタとも呼ばれる小さな行列です。その出力が特徴マップで、フィルタが反応した場所ほど大きな値になります。サポートベクターマシンのカーネルとは全くの別物です。

### 確認問題 第5部

#### 第3問 畳み込みニューラルネットワークにおいて、パディングの効果の説明として、最も適切な選択肢を1つ選べ

答

**出力の大きさを調整でき、端の情報が失われるのを防げる**

パディングは入力の周りを0などで埋めてから畳み込みます。埋めないと端の画素が中心になる機会が少なく、また出力が入力より小さくなります。大きさをそろえたいときにも使われます。

## 60語チェックリスト

自分の言葉で説明できたら✓を入れてください。読めるだけでは足りません。

- |                                                          |                                                |                                     |
|----------------------------------------------------------|------------------------------------------------|-------------------------------------|
| <input type="checkbox"/> 重み                              | <input type="checkbox"/> 特徴マップ                 | <input type="checkbox"/> 平均値プーリング   |
| <input type="checkbox"/> 線形関数                            | <input type="checkbox"/> パディング                 | <input type="checkbox"/> ResNet     |
| <input type="checkbox"/> Atrous Convolution              | <input type="checkbox"/> フィルタ                  | <input type="checkbox"/> BPTT       |
| <input type="checkbox"/> Depthwise Separable Convolution | <input type="checkbox"/> グループ正規化               | <input type="checkbox"/> GRU        |
| <input type="checkbox"/> Dilated Convolution             | <input type="checkbox"/> バッチ正規化                | <input type="checkbox"/> LSTM       |
| <input type="checkbox"/> カーネル                            | <input type="checkbox"/> レイヤー正規化               | <input type="checkbox"/> エルマンネットワーク |
| <input type="checkbox"/> スライド                            | <input type="checkbox"/> インスタンス正規化             | <input type="checkbox"/> 勾配消失問題     |
| <input type="checkbox"/> 畳み込み操作                          | <input type="checkbox"/> グローバルアベレージプーリング (GAP) | <input type="checkbox"/> 勾配爆発問題     |
| <input type="checkbox"/> 畳み込みニューラルネットワーク (CNN)           | <input type="checkbox"/> 最大値プーリング              | <input type="checkbox"/> 教師強制       |
|                                                          | <input type="checkbox"/> 不変性の獲得                | <input type="checkbox"/> ゲート機構      |

- |                                                     |                                                        |                                         |
|-----------------------------------------------------|--------------------------------------------------------|-----------------------------------------|
| <input type="checkbox"/> 双方向RNN (Bidirectional RNN) | <input type="checkbox"/> 位置エンコーディング                    | <input type="checkbox"/> Crop           |
| <input type="checkbox"/> 時系列データ                     | <input type="checkbox"/> キー                            | <input type="checkbox"/> CutMix         |
| <input type="checkbox"/> ジョルダンネットワーク                | <input type="checkbox"/> クエリ                           | <input type="checkbox"/> Cutout         |
| <input type="checkbox"/> リカレントニューラルネットワーク(RNN)      | <input type="checkbox"/> バリュー                          | <input type="checkbox"/> Mixup          |
| <input type="checkbox"/> Attention                  | <input type="checkbox"/> VQ-VAE・info VAE・ $\beta$ -VAE | <input type="checkbox"/> noising        |
| <input type="checkbox"/> Multi-Head Attention       | <input type="checkbox"/> 次元削減                          | <input type="checkbox"/> paraphrasing   |
| <input type="checkbox"/> Self-Attention             | <input type="checkbox"/> 事前学習                          | <input type="checkbox"/> RandAugment    |
| <input type="checkbox"/> Seq2Seq                    | <input type="checkbox"/> 積層オートエンコーダ                    | <input type="checkbox"/> Random Erasing |
| <input type="checkbox"/> Source-Target Attention    | <input type="checkbox"/> 変分オートエンコーダ(VAE)               | <input type="checkbox"/> Random Flip    |
| <input type="checkbox"/> Transformer                | <input type="checkbox"/> Contrast                      | <input type="checkbox"/> Rotate         |
|                                                     | <input type="checkbox"/> Brightness                    |                                         |

## 第6部

# ディープラーニングの応用例

技術分野／9の中項目・131語

- 26 画像認識 (31語)
- 27 自然言語処理 (25語)
- 28 音声処理 (17語)
- 29 深層強化学習 (20語)
- 30 データ生成 (9語)
- 31 転移学習・ファインチューニング (7語)
- 32 マルチモーダル (10語)
- 33 モデルの解釈性 (6語)
- 34 モデルの軽量化 (6語)

## 26 画像認識

→ **この節の目標。**画像認識タスクの種類とその概要について／代表的な画像認識モデルについて／画像認識が実世界において、どのように活用されているか。

### AlexNet

2012年のILSVRCで優勝した畳み込みニューラルネットワークです。トロント大学のヒントンのチームが作りました。ReLU関数、ドロップアウト、GPUによる学習を組み合わせ、二位に大差をつけて第三次AIブームの引き金になりました。

### DeepLab

セマンティックセグメンテーションのためのモデルです。フィルタに隙間を空けるAtrous Convolutionを使い、パラメータを増やさずに広い範囲の文脈を捉えます。画素ごとの塗り分けの精度を高めました。

### DenseNet

各層の出力を、それ以降のすべての層へつなぐ構造を持つネットワークです。前の層で得た特徴を何度も使い回せるため、少ないパラメータでも高い性能が出ます。ResNetのスキップ結合を、さらに推し進めた形と言えます。

### EfficientNet

ネットワークの深さ、幅、入力画像の大きさの三つを、決まった比率でまとめて拡大する設計のモデルです。これを複合スケージングと呼びます。どれか一つだけを大きくするより効率がよく、少ない計算量で高い精度を出します。

### Fast R-CNN

物体検出のモデルで、R-CNNを高速にしたものです。領域ごとに何度も畳み込むのではなく、画像全体を一度だけ畳み込み、その結果から領域ごとの特徴を切り出します。この工夫で処理時間が大きく縮みました。

### Faster R-CNN

Fast R-CNNをさらに速くしたモデルです。どこに物体がありそうかという領域の候補づくりも、ネットワークの中で行うようにしました。全体を一つのネットワークとして学習でき、精度と速度の両方が向上しています。

### FCN Fully Convolutional Network

全結合層をなくし、畳み込み層だけで構成したネットワークです。出力が二次元のまま保たれるため、画素ごとにどのカテゴリかを判定できます。セマンティックセグメンテーションの先駆けとなったモデルです。

## FPN

解像度の異なる複数の特徴マップを組み合わせる構造です。浅い層の細かい情報と、深い層の意味的な情報を両方使えます。これによって、大きな物体も小さな物体も同じように検出できるようになります。

## GoogLeNet

2014年のILSVRCで優勝したモデルです。大きさの違う複数のフィルタを並列に適用するInceptionモジュールを積み重ねる構造が特徴です。層を深くしながらも、パラメータの数を抑える工夫が施されています。

## Mask R-CNN

Faster R-CNNに、物体ごとの領域を塗り分ける枝を追加したモデルです。物体の位置を四角い枠で示すだけでなく、その形まで求められます。インスタンスセグメンテーションを実現する代表的な手法です。

## MnasNet

ネットワークの構造を自動で探索するNASを使い、さらにスマートフォン上での実際の処理速度も評価に加えて設計されたモデルです。計算量という理論値ではなく、実機での速さを直接目指した点が特徴です。

## MobileNet

スマートフォンなど、計算資源の限られた機器で動かすために作られた軽量なモデルです。畳み込みを空間方向とチャンネル方向に分けるDepthwise Separable Convolutionによって、計算量を大幅に減らしています。

## NAS Neural Architecture Search

ネットワークの構造そのものを、人ではなく機械が自動で探索する技術です。層の並べ方やフィルタの大きさなどを試行錯誤させて、良い構造を見つけます。設計者の勘に頼らずに済む一方、探索に膨大な計算が必要です。

## Open Pose

画像から人の姿勢を推定するモデルです。複数の人が写っていても、それぞれの関節の位置と、どの関節同士がつながっているかを同時に求められます。姿勢推定の代表的な手法として挙げられます。

## PSPNet

セマンティックセグメンテーションのモデルです。複数の大きさにプーリングした結果を組み合わせることで、画像全体の文脈を捉えます。局所しか見ないと起きる取り違えを、広い範囲の情報で防ぐ狙いがあります。

## ResNet

層を飛び越えて入力を足すスキップ結合によって、非常に深いネットワークの学習を可能にしたモデルです。2015年のILSVRCで優勝しました。以降の多くのモデルが、この構造を土台として取り入れています。

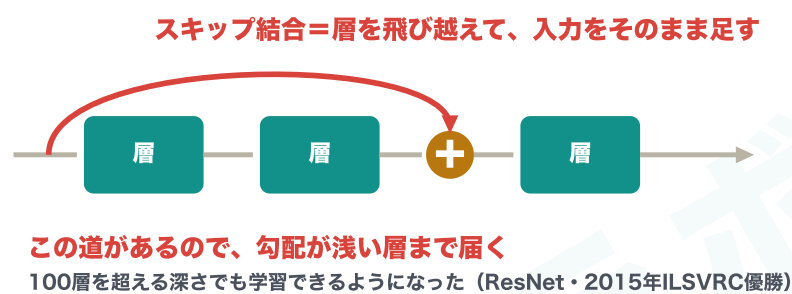


図 ResNet

## SegNet

エンコーダで小さくし、デコーダで元の大きさへ戻す構造のセマンティックセグメンテーションのモデルです。プーリングでどこが最大だったかを記憶し、戻すときにその位置情報を使う点が特徴です。

## SENet

チャンネルごとの重要度を学習し、重み付けする仕組みを加えたモデルです。全体を一度圧縮してから各チャンネルの重みを求める、Squeeze-and-Excitationという構造を持ちます。2017年のILSVRCで優勝しました。

ILSVRC（画像認識のコンテスト）で優勝したモデル



**2012年のAlexNetが第三次AIブームの引き金になった**

図 SENet

## SSD

領域の候補をあらかじめ作らず、一度の処理で検出まで済ませる物体検出モデルです。解像度の異なる複数の特徴マップを使うため、大きささまざまな物体に対応できます。YOLOと同じ、一段階方式に分類されます。

## U-Net

エンコーダとデコーダをU字型に配置し、対応する層どうしを直接つないだモデルです。細かい位置の情報が失われにくく、少ないデータでも学習できます。医療画像のセグメンテーションで広く使われています。

## VGG

3×3という小さなフィルタだけを積み重ねた、単純で分かりやすい構造のモデルです。小さいフィルタを重ねれば大きなフィルタと同じ範囲を、より少ないパラメータで見られます。層を深くすることの効果を示しました。

## Vision Transformer

画像を小さな区画に分割し、それを並んだ要素とみなしてTransformerで処理するモデルです。畳み込みを一切使いません。大量のデータで学習させれば、畳み込みを使ったモデルを上回る性能が出ることを示しました。

## Wide ResNet

ResNetを深くする代わりに、各層のチャンネル数、つまり幅を広げたモデルです。深さだけが性能を決めるわけではないことを示しました。層数が少ないぶん、並列に計算しやすく学習が速いという利点もあります。

## YOLO

画像を一度見るだけで物体の位置と種類を同時に求める検出モデルです。名前はYou Only Look Onceの略で、その通り一段階で処理します。非常に高速なため、動画のリアルタイム処理に向いています。

## 一般物体認識

あらかじめ限定された対象だけでなく、日常にある一般的な物体を広く認識することです。特定の一種類だけを見分ける特定物体認識と対比されます。ILSVRCが対象としていたのも、この一般物体認識です。

## インスタンスセグメンテーション

画素ごとに塗り分けただけで、同じ種類の物体でも一つひとつを別のものとして区別する手法です。三人写っていれば三人を別々に塗り分けます。個体を区別しないセマンティックセグメンテーションとの違いが問われます。

## 姿勢推定

画像や動画から、人の関節の位置を推定する技術です。肩、肘、手首といった点を求め、それらのつながりから体の姿勢を把握します。スポーツの動作解析や、見守りシステムなどに使われています。

## セマンティックセグメンテーション

画像の画素一つひとつが、どのカテゴリに属するかを判定して塗り分ける手法です。ただし同じ種類の物体は区別せず、三人写っていても全部まとめて人として塗ります。個体を区別するインスタンス版と対にします。

## 物体検出

画像の中で物体がどこにあるかを四角い枠で示し、あわせてその種類も判定する技術です。位置と種類の両方を求める点が、種類だけを答える物体識別との違いです。YOLOやSSD、R-CNNの系統が代表です。

## 物体識別

画像に何が写っているかという、種類だけを判定する技術です。どこにあるかという位置は求めません。画像分類とも呼ばれ、画像認識の中で最も基本的な課題にあたります。物体検出と混同しないよう区別します。

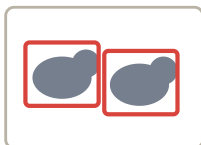
## パノプティックセグメンテーション

セマンティックセグメンテーションとインスタンスセグメンテーションを統合した手法です。空や道路のような数えられないものはカテゴリで塗り、人や車のような数えられるものは個体ごとに区別します。

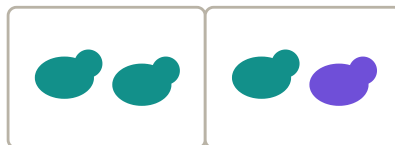
**物体識別**  
何が写っているか



**物体検出**  
+どこにあるか



**セマンティック**   **インスタンス**  
画素ごとに塗る   +個体も区別



**セマンティックは種類でまとめて塗る／インスタンスは1匹ずつ分ける**  
この2つを合わせたものがパノプティックセグメンテーション

図 パノプティックセグメンテーション

## 27 自然言語処理

→ **この節の目標。** 自然言語処理タスクの種類とその概要について／自然言語処理タスクにおける特徴表現とその手法について／代表的な自然言語処理モデルについて／自然言語処理が実世界において、どのように活用されているか。

### BERT

2018年にGoogleが発表した言語モデルです。Transformerのエンコーダを使い、文の前と後ろの両方向から文脈を読みます。文中の一部を隠して当てさせる学習によって、多くの課題で当時の最高性能を出しました。

### BoW Bag-of-Words

文書を、語がそれぞれ何回出てきたかという回数だけで表す方法です。袋に語を放り込んだ状態にたとえた名前で、語順の情報は完全に失われます。単純ですが、文書分類などでは今も基準として使われます。

### CBOW

word2vecの学習方式の一つです。周りにある語から、真ん中の語が何かを当てさせることで、語のベクトルを学習します。中心の語から周りを当てるスキップグラムとは、向きが逆になります。

### CEC

LSTMの内部で、情報をそのまま保ち続ける記憶セルのことです。ここを通る経路では値が掛け合わされずに流れるため、勾配が小さくなりません。LSTMが長期の依存を扱える、中心的な仕組みです。

### ELMo

文脈に応じて、同じ語でも違うベクトルを与える手法です。双方向のLSTMを使って前後の文脈を読み取ります。word2vecでは一つの語に一つのベクトルしか与えられなかった、という限界を越えました。

### fastText

word2vecを発展させた手法です。語をそのまま扱うのではなく、文字のまとまりに分解して学習します。そのため、学習時になかった未知の語でもベクトルを作れます。活用形の多い言語に強いという利点もあります。

### GLUE

自然言語を理解する能力を測るための、複数の課題をまとめたベンチマークです。文の含意関係の判定や感情分析などが含まれます。モデルどうしを共通のものさしで比べられるようにした点に意味があります。

## n-gram

連続するn個の語、あるいは文字のまとまりを指します。二つならバイグラム、三つならトライグラムです。前のいくつかの語から次の語を予測するという、統計的な言語モデルの基本的な考え方になっています。

## PaLM

Googleが開発した大規模言語モデルです。膨大なパラメータを持ち、推論を要する課題でも高い性能を示しました。規模を大きくすることで能力が飛躍的に伸びる、という流れを代表するモデルの一つです。

## Seq2Seq

入力の系列をエンコーダでまとめ、デコーダで別の系列に変換する枠組みです。機械翻訳や文書要約に使われました。入力を固定長にまとめる点が長い文で弱点となり、Attentionが生まれるきっかけになりました。

## TF-IDF

語の重要度を測る指標です。その文書の中でよく出てくる語ほど重要とし、同時に、どの文書にも出てくるありふれた語ほど重要度を下げます。この組み合わせによって、その文書らしい語を取り出せます。

## word2vec

語を、意味の近さが計算できるベクトルに変換する手法です。似た文脈で使われる語は似たベクトルになります。ベクトルの引き算と足し算で意味の関係を表せることが知られ、分散表現を一気に広めました。

## 感情分析

音声から、話し手の感情を読み取る課題です。声の高さ、大きさ、話す速さといった特徴を手がかりにします。同じ言葉でも言い方で意味が変わるため、文章だけからは分からない情報が取れる点に価値があります。

## 機械翻訳

ある言語の文を別の言語へ自動で翻訳する課題です。人が規則を書くルールベース、対訳データから確からしさを学ぶ統計的機械翻訳、そしてニューラルネットワークを使う方式へと移り変わってきました。

## 形態素解析

文を、意味を持つ最小の単位である形態素に分け、それぞれの品詞を判定する処理です。英語と違って日本語は語の間に空白がないため、この処理が自然言語処理の最初の一步として欠かせません。MeCabなどの道具が使われます。

## 構文解析

文の中で、どの語がどの語に係っているかという関係を明らかにする処理です。形態素解析で切り分けた語どうしを、木の形の構造にまとめます。同じ並びでも係り方が二通りある文もあり、意味を正しく取る土台になります。

## 質問応答

自然な文で書かれた質問に対して、答えを返す課題です。関係する文書を並べる情報検索とは違い、答えそのものを示します。IBMのワトソンが代表例で、現在の対話型AIにもつながっています。

## 情報検索

大量の文書の中から、求める内容に関するものを探し出す課題です。検索エンジンが代表例です。語の重要度を測るTF-IDFなどが古くから使われ、近年は分散表現を用いた意味による検索も広がっています。

## スキップグラム

word2vecの学習方式の一つです。真ん中の語から、その周りに現れる語を当てさせることでベクトルを学習します。周りから中心を当てるCBOWとは向きが逆で、まれにしか現れない語に強いとされます。

## 単語埋め込み

語を、低い次元の実数のベクトルに置き換えて表すことです。ワンホットベクトルのように次元が語彙数まで膨らむことがなく、語どうしの意味の近さも計算できます。分散表現とほぼ同じ意味で使われます。

## 分散表現

語の意味を、多数の次元に分け持たせて表す方法です。一つの次元が一つの意味を担うのではなく、全体の組み合わせで意味を表します。ワンホットベクトルと違い、語どうしの近さを距離として測れるのが利点です。

**ワンホットベクトル**

語彙の数だけ次元／どの語どうしも同じ距離



0 0 1 0 0 0 0 ← 「犬」はここだけ1

**分散表現（単語埋め込み）**

低い次元／意味の近さが距離になる

**王 - 男 + 女 ≒ 女王**

意味の関係が、ベクトルの足し引きで表せる

図 分散表現

**文書要約**

長い文章を短くまとめる課題です。元の文をそのまま抜き出す抽出型と、新しく文を作り直す生成型に分かれます。生成型は自然に読める一方で、元の文書にない内容を作ってしまう危険があり、事実確認が欠かせません。

**ワンホットベクトル**

語彙の数だけ次元を用意し、その語に当たる場所だけを1、残りをすべて0とする表し方です。作り方は簡単ですが、語彙が増えるほど次元が膨らみ、しかもどの語どうしも同じ距離になるため意味の近さを表せません。

**大規模言語モデル LLM**

膨大な量のテキストで学習した、非常に大きな言語モデルです。パラメータの数とデータ量を増やすほど性能が伸びることが分かり、規模の拡大が進みました。文章生成や要約、翻訳などを一つのモデルでこなします。

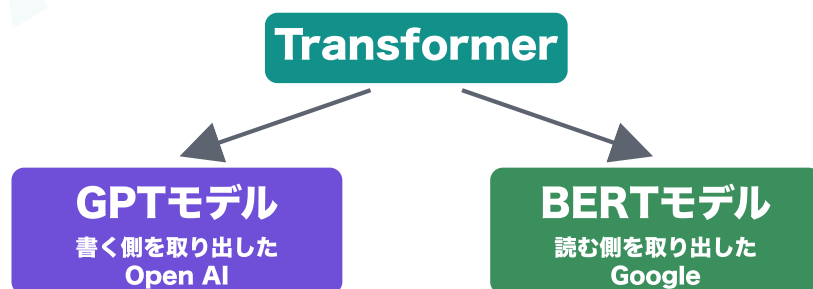


図 大規模言語モデル(LLM)

**統計的機械翻訳**

大量の対訳データから、訳し方の確からしさを統計的に学んで翻訳する方式です。人が規則を書く方式に代わって主流となりました。現在はニューラルネットワークを用いる方式に置き換わっています。

## 28 音声処理

→ **この節の目標。** 音声処理タスクの種類とその概要について／音声処理タスクにおける特徴表現とその手法について／代表的な音声処理モデルについて／音声処理が実世界において、どのように活用されているか。

### A-D変換

アナログの音の波を、コンピュータが扱える数値に変換することです。一定の間隔で値を拾う標本化、値を段階に丸める量子化、それを数値として記録する符号化の三段階で行われます。音声処理の入り口にあたります。

### WaveNet

DeepMindが開発した音声合成のモデルです。音の波形そのものを、一点ずつ順に直接作り出します。それまでの、録音した断片をつなぎ合わせる方式に比べ、はるかに自然で滑らかな音声を実現しました。

### 音韻

ある言語の中で、意味の違いを生み分ける音の単位です。実際に発せられた物理的な音そのものではなく、その言語の話し手が同じ音だと認識するまとまりを指します。物理的な単位である音素と、対にして押さえます。

### 音声合成

文字で書かれた内容から、人が話しているような音声を作り出す技術です。読み上げソフトやカーナビの案内に使われます。近年はWaveNetのように波形を直接生成する方式で、自然さが大きく向上しました。

### 音声認識

人の話した音声を、文字に変換する技術です。かつては隠れマルコフモデルが主力でしたが、現在はディープラーニングが用いられます。スマートスピーカーや自動字幕など、身近な用途が広がっています。

### 音素

言語の音を分ける、最小の単位です。日本語であれば、母音と子音の組み合わせがこれに当たります。音声認識では、まず音素を推定し、それをつないで語や文にするという段階を踏むことがあります。

### 隠れマルコフモデル

直接は観測できない状態の移り変わりを、確率で表すモデルです。音声のように、時間とともに変化する系列を扱えます。ディープラーニングが普及する以前、音声認識の主力として長く使われてきました。

## 感情分析

音声から、話し手の感情を読み取る課題です。声の高さ、大きさ、話す速さといった特徴を手がかりにします。同じ言葉でも言い方で意味が変わるため、文章だけでは分からない情報が取れる点に価値があります。

## 高速フーリエ変換 FFT

波形を、どんな周波数の成分がどれだけ含まれているかに分解する計算を、高速に行う方法です。音声はそのままでは扱いにくいので、周波数の情報に直してから特徴を取り出すのが基本の流れです。

## スペクトル包絡

周波数ごとの強さを並べたとき、その山の連なりをなめらかにつないだ線のことです。声道の形によって決まるため、どんな音を発しているかを表します。声の高さとは分けて捉えられるのが利点です。

## パルス符号変調器 PCM

アナログの音をデジタルに変換する、代表的な方式です。標本化、量子化、符号化という三つの段階を経て数値に直します。音楽CDなどにも使われている最も基本的な記録方式で、A-D変換の具体例にあたります。

## フォルマント

スペクトル包絡に現れる、いくつかの山のことで、声道の共鳴によって、特定の周波数が強められることで生じます。この山の位置が、どの母音を発しているかを決める重要な手がかりになります。

## フォルマント周波数

フォルマント、つまりスペクトル包絡の山が現れる周波数のことです。低い方から順に、第一フォルマント、第二フォルマントと呼びます。この一番目と二番目の組み合わせを見ることで、どの母音かを見分けられます。

## メル周波数ケプストラム係数 MFCC

人の耳の聞こえ方に合わせて、音声の特徴を数値の並びに直したものです。声の高さの影響を抑えつつ、どんな音かという情報を取り出せます。ディープラーニング以前から、音声認識の特徴量として広く使われてきました。

## 音声認識までの流れ



**そのままでは扱えないので、周波数に直してから特徴を取り出す**

かつては隠れマルコフモデル、いまはディープラーニングが使われる

図 メル周波数ケプストラム係数(MFCC)

## メル尺度

人が実際に感じる音の高さに合わせて作られた尺度です。人の耳は低い音の違いには敏感で、高い音の違いには鈍いという性質があります。その偏りを反映させた尺度で、MFCCを求める際に使われます。

## 話者識別

話しているのが誰かを、声から判定する技術です。何を話したかを問う音声認識とは目的が異なります。本人確認としての利用が広がる一方、生成AIで声を似せる技術も進んでおり、扱いに注意が要る分野です。

## CTC

入力と出力の長さが違う系列を、どこがどこに対応するかを人が与えなくても学習できるようにする損失関数です。音声とその書き起こしのように、長さも区切りも違うものを結びつけられます。音声認識で広く使われます。

## 29 深層強化学習

⇒ **この節の目標。** 代表的な強化学習モデルについて／強化学習が実世界において、どのように活用されているか。

### A3C

複数のエージェントを並列に動かし、それぞれの経験をまとめて学習する手法です。同時に多様な経験を集められるため、学習が速く、かつ安定します。深層強化学習を実用的にした手法の一つです。

### Agent57

DeepMindが開発したエージェントで、Atariの57種類のゲームすべてで人間の水準を超えました。それまで苦手とされてきた、報酬がなかなか得られない種類のゲームも攻略した点が大きな成果です。

## APE-X

経験を集める役割と、そこから学習する役割を分けて、大規模に並列化した手法です。多数のエージェントが集めた経験を一か所にため、学習側がそこから取り出して使います。学習の効率を大きく高めました。

## DQN

Q学習にディープラーニングを組み合わせた手法です。DeepMindがAtariのゲームで人間並みの成績を出し、深層強化学習の出発点となりました。経験を貯めて後から使う経験再生などの工夫で、学習を安定させています。

## OpenAI Five

OpenAIが開発した、Dota 2というゲームのエージェントです。五人一組で戦う複雑なゲームで、人間のプロチームに勝ちました。複数のエージェントが協力する場面での成果として知られます。

## PPO

方策を一度にどれだけ変えてよいかに制限を設けることで、学習を安定させた方策勾配法です。実装が比較的簡単で性能も安定するため、深層強化学習の標準的な手法として広く使われています。RLHFでも用いられます。

## Rainbow

DQNに対して提案されてきた改良手法を、七つまとめて組み合わせたものです。ダブルDQNやデュエリングネットワーク、ノイジーネットワークなどが含まれます。組み合わせることで、単独よりも高い性能を出しました。

## RLHF

人間が示した好みを報酬として学習させる手法です。どちらの出力が良いかを人が選び、その判断を再現する報酬モデルを作って、それを使って調整します。ChatGPTを人にとって好ましい応答へ導いた仕組みです。



**RLHF=人間のフィードバックによる強化学習**

図 RLHF

## sim2real

シミュレータの中で学習した方策を、現実の機械へ移すことです。現実で試すのは費用も危険も大きい  
ため、まず仮想空間で学ばせます。ただし現実とのずれが必ず生じるため、その差をどう埋めるかが課  
題になります。

## アルファスター AlphaStar

DeepMindが開発した、StarCraft IIというゲームのエージェントです。盤面の一部しか見えず、行動の  
選択肢も膨大という難しい条件で、トップ級の人間プレイヤーに勝ちました。

## オフライン強化学習

あらかじめ集めておいたデータだけを使い、環境と実際にやりとりせずに学習する方法です。試行錯誤  
が危険だったり費用がかかったりする医療や自動運転で期待されています。データにない行動を評価で  
きない点が難所です。

## 残差強化学習

すでにある制御の仕組みを土台として残し、それとの差分だけを強化学習で学ぶ方法です。ゼロから学  
ばせるより早く、失敗も少なくて済みます。既存の技術と強化学習を組み合わせる、現実的なやり方  
です。

## 状態表現学習

カメラの映像のような生の観測から、意思決定に役立つ形の表現を学び取ることです。そのままでは情  
報が多すぎて学習が進まないため、意味のある特徴に圧縮します。強化学習を現実の問題に使うための  
鍵になります。

## ダブルDQN

どの行動を選ぶかと、その行動の価値をいくらと見るかを、別々のネットワークで行う手法です。  
DQNでは価値を高く見積もりすぎる偏りがありましたが、役割を分けることでその偏りを抑えられま  
す。

## デュエリングネットワーク

状態そのものの良さと、その状態でどの行動が優れているかという二つを、分けて学習する構造です。  
どの行動を選んででも大差ない場面でも、状態の価値だけは正しく学べるようになり、学習の効率が上が  
ります。

## ドメインランダムマイゼーション

シミュレータの見た目や物理の条件をランダムに変えながら学習させる方法です。さまざまな条件を経験させることで、現実の条件がそのどこかに収まり、そのまま通用しやすくなります。sim2realの代表的な対策です。

## ノイズネットワーク

ネットワークの重みそのものに雑音を加えることで、行動にばらつきを生ませる手法です。 $\epsilon$ の確率でランダムに動く $\epsilon$ -greedy方策の代わりになります。どれだけ探索するかも学習によって決められる点が利点です。

## 報酬成形

学習が進みやすいように、途中の段階でも報酬を与えるよう人が設計することです。最終的な成功だけを報酬にすると、そこへたどり着けず学習が進みません。ただし設計を誤ると、意図しない抜け道を学んでしまいます。

## マルチエージェント強化学習 MARL

複数のエージェントが同じ環境の中で、同時に学習する枠組みです。互いの行動が環境を変えるため、一人で学ぶ場合より難しくなります。協力する場合と、競い合う場合の両方が研究されています。

## 連続値制御

行動の選択肢が、飛び飛びの選択肢ではなく連続した値になる問題です。ロボットの関節をどの角度まで動かすか、といった場合が当たります。価値を比べるQ学習系より、方策勾配法の方が向いています。

## 30 データ生成

⇒ **この節の目標。** データ生成タスクの種類とその概要について／代表的なデータ生成モデルについて／データ生成モデルが実世界において、どのように活用されているか。

## CycleGAN

対になっていない画像どうしても、変換の仕方を学習できるGANです。馬の画像とシマウマの画像を、一枚ずつ対応づけなくても互いに変換できます。変換して戻したら元に戻るという条件を課すことで実現しています。

## DCGAN

GANに畳み込みニューラルネットワークを取り入れたモデルです。学習が不安定になりやすいGANに対し、構造上の工夫を加えることで安定させました。GANが実用的に使われるようになる出発点となりました。

## Diffusion Model

雑音だらけの状態から、少しずつ雑音を取り除いていくことで画像を作り出すモデルです。学習の際は逆に、画像へ徐々に雑音を加える過程を学びます。現在の画像生成の主流であり、品質と安定性でGANを上回りました。

## NeRF

いくつかの方向から撮った写真をもとに、その場面を三次元として再現し、撮っていない角度からの見え方まで作り出す手法です。空間のある点をどの方向から見たらどう見えるかを、ネットワークに学習させます。

## Pix2Pix

対になった画像の組から、変換の仕方を学習するGANです。線画から写真、白黒から着色、といった変換ができます。対応する画像の組を用意する必要がある点が、CycleGANとの違いです。

## 音声生成

音声を人工的に作り出す技術です。文章を読み上げる音声合成のほか、特定の人の声に似せることもできます。WaveNetのように波形を直接作る方式によって自然さが向上した一方、なりすましへの懸念も生まれています。

## 画像生成

学習していない新しい画像を作り出す技術です。GANや変分オートエンコーダ、そして拡散モデルによって実現されます。近年は文章から画像を作る形が広まり、生成AIの代表的な用途になっています。

## 敵対的生成ネットワーク GAN

本物らしいデータを作る生成器と、本物か偽物かを見分ける識別器を競い合わせて学習させる仕組みです。2014年にイアン・グッドフェローが提案しました。互いに鍛え合うことで、生成の質が高まっていきます。

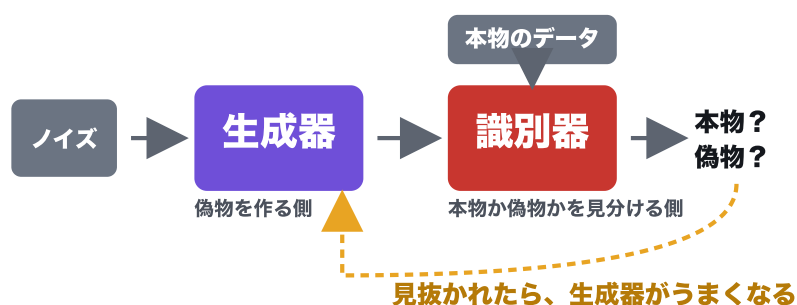


図 敵対的生成ネットワーク (GAN)

## 文章生成

文章を人工的に作り出す技術です。大規模言語モデルによって、自然で長い文章が作れるようになりました。一方で、事実と異なる内容をもっともらしく書いてしまうハルシネーションという問題も抱えています。

## 31 転移学習・ファインチューニング

⇒ **この節の目標。** 転移学習とファインチューニングの基礎的な知識／どのような場面やタスクにおいて、転移学習・ファインチューニングが効果を発揮するのか／転移学習・ファインチューニングによって、様々なタスクにおいて大幅な精度向上を果たした代表的なモデルについて。

### Few-shot

ごく少数の例だけを見せて、新しい課題に対応させることです。大規模言語モデルでは、指示文の中に例を数個入れるだけで、そのやり方に沿った出力が得られます。学習し直さずに済む点が大きな利点です。

### One-shot

例をたった一つだけ見せて、新しい課題に対応させることです。Few-shotの中でも、特に例が少ない場合を指します。例を全く与えないZero-shot、数個与えるFew-shotと合わせて、三段階で対比して覚えます。

### 自己教師あり学習

データそのものから正解を作り出して学習する方法です。文章の一部を隠して当てさせる、画像の一部を隠して復元させる、といった形を取ります。人手でラベルを付ける必要がなく、大量のデータを活かせます。

### 事前学習

目的の課題に取りかかる前に、大量のデータで先に学習しておくことです。言葉や画像の一般的な性質を身につけた状態から始められるため、その後の学習が少ないデータで済みます。基盤モデルの土台となる考え方です。

### 事前学習済みモデル

すでに大量のデータで学習を終えたモデルのことです。これを出発点にして、自分の目的に合わせて調整することを、ファインチューニングと呼びます。学習にかかる時間とデータを大きく節約できます。

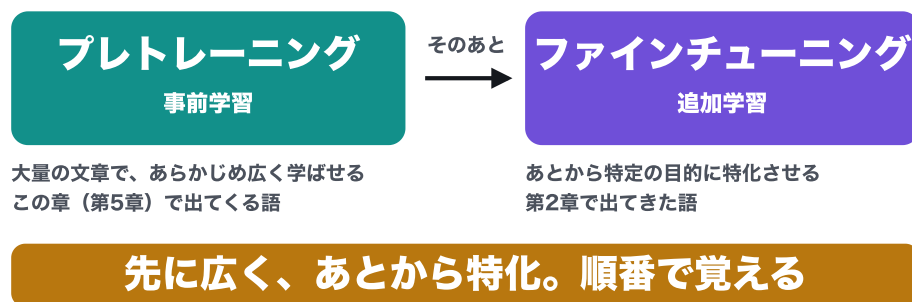


図 事前学習済みモデル

## 破滅的忘却

新しい課題を学習させると、それ以前に学んだ内容を大きく失ってしまう現象です。人間のように少しずつ積み上げることが難しい、という限界を表しています。継続的に学び続けさせるうえでの、大きな課題です。

## 半教師あり学習

少量の正解付きデータと、大量の正解なしデータを組み合わせて学習する方法です。正解を付ける作業には手間も費用もかかるため、その負担を抑えつつ精度を高められます。教師ありと教師なしの中間に位置します。

## 32 マルチモーダル

→ **この節の目標。** マルチモーダルタスクの種類とその概要について／代表的なマルチモーダルモデルについて／マルチモーダルモデルが実世界において、どのように活用されているか。

### CLIP

画像と、それを説明する文章を、同じ空間の中で近い位置に置くように学習したモデルです。OpenAIが開発しました。学習していない分類でも、文章で条件を書くだけで判定できるようになります。

### DALL-E

文章の指示から画像を作り出すモデルです。OpenAIが発表し、テキストから画像を生成するという用途を一般に広めました。現実には存在しない組み合わせの絵も作れる点が、大きな驚きを呼びました。

### Flamingo

画像と文章が交互に混ざった入力を、そのまま扱える視覚言語モデルです。少数の例を示すだけで、学習し直さずに新しい課題へ対応できます。画像について会話するという使い方の、先駆けとなったモデルです。

## Image Captioning

画像を見て、その内容を説明する文章を作る課題です。画像を理解する力と、文章を組み立てる力の両方が必要になります。視覚に障害のある方への読み上げや、大量の画像に説明を付ける作業の自動化などに使われています。

## Text-To-Image

文章の指示から画像を作り出す課題です。DALL-EやStable Diffusionが代表例で、生成AIが一般に広まる大きなきっかけになりました。学習データの著作権をめぐる議論も、ここから起きています。

## Visual Question Answering

画像を見せたうえで、その画像についての質問に答える課題です。写真に何人写っているか、といった問いに答えます。画像の理解と言語の理解を同時に必要とする、代表的なマルチモーダルの課題です。

## Unified-IO

画像、文章、領域の指定といった多様な入力と出力を、一つのモデルでまとめて扱えるようにしたモデルです。課題ごとにモデルを作り分けるのではなく、統一した形で解こうとする流れを代表します。

## Zero-shot

例を一つも与えずに、指示だけで新しい課題をこなすことです。事前学習で身につけた広い知識があるからこそ成り立ちます。例を数個示すFew-shot、一つだけ示すOne-shotと並べて押さえます。

### Zero-Shot プロンプティング

見せる例

な し

いきなり頼む

### Few-Shot プロンプティング

見せる例

例①

例②

例③

見せてから頼む

**Shot は「回数」ではなく「例の数」**

図 Zero-shot

## 基盤モデル

大量で多様なデータを使って学習し、さまざまな用途に適応させられる大規模なモデルです。一つの土台から、翻訳にも要約にも画像生成にも展開できます。現在の生成AIは、ほぼこの考え方の上に立っています。

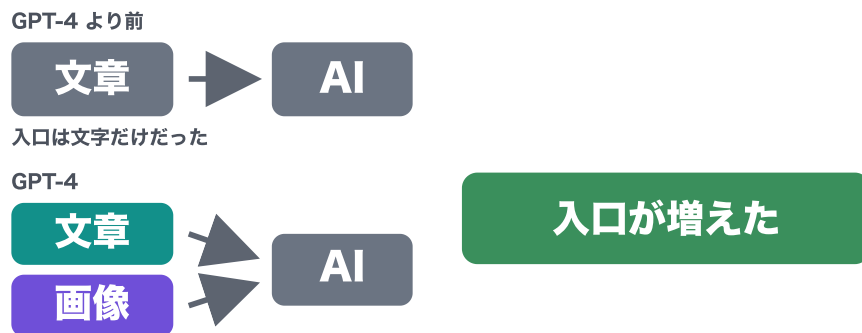


図 基盤モデル

## マルチタスク学習

複数の課題を、一つのモデルで同時に学習させる方法です。課題どうしで共通する知識を分け合えるため、それぞれ単独で学ぶより性能が上がる場合があります。学習データが少ない課題ほど恩恵を受けます。

## 33 モデルの解釈性

⇒ **この節の目標。**モデルの解釈性が必要な背景について／解釈性が必要なユースケースについて／解釈性の向上に寄与する代表的な手法について。

### CAM

畳み込みニューラルネットワークが、画像のどこを見て判断したのかを可視化する手法です。グローバルアベレージプリーングの仕組みを利用して、判断の根拠となった場所を色の濃淡で示します。構造に制約がある点が弱点です。

### Grad-CAM

勾配の情報を使って、判断の根拠となった場所を可視化する手法です。CAMと違い、ネットワークの構造を変えなくても使えるため、適用できる範囲が広がっています。説明可能AIの代表的な手法です。

### LIME

ある予測の周辺だけを、単純で分かりやすいモデルで近似して説明する手法です。中身の分からないモデルにもそのまま使えます。ただし説明はあくまでその一件の周辺に限られる点に注意が必要です。

### Permutation Importance

ある特徴の値だけをでたらめに並べ替え、そのとき精度がどれだけ落ちるかで、その特徴の重要さを測る手法です。大きく落ちるほど重要だったと分かります。モデルの中身によらず使えるのが利点です。

## SHAP

それぞれの特徴が予測にどれだけ寄与したかを、協力ゲーム理論のシャープレイ値を使って公平に配分する手法です。理論的な裏づけがあり、一件ごとの説明にも全体の傾向にも使えるため、広く用いられています。

## 説明可能AI XAI

なぜその判断に至ったのかを、人が理解できる形で示せるようにする研究分野です。医療や与信のように、判断の理由が問われる場面では欠かせません。性能の高さと説明のしやすさは、しばしば両立が難しくなります。

説明可能AI (XAI) =なぜその判断をしたかを示す

### CAM / Grad-CAM

画像のどこを見たか  
CNNの中を使う

### LIME

その1件の周りを単純化  
中身を知らなくてよい

### SHAP

各特徴の寄与を公平に配分  
理論の裏づけがある

**Permutation Importanceは、値を並べ替えて精度の落ち方を見る**

性能の高さと説明のしやすさは、しばしば両立しない

図 説明可能AI (XAI)

## 34 モデルの軽量化

⇒ **この節の目標。**モデルの軽量化が必要な背景について／モデルの軽量化が必要なユースケースについて／代表的なモデル軽量化手法について。

## エッジAI

手元の機器の中でAIを動かすことです。撮影した映像などを外へ送らずに済むため、通信が要らず、応答も速く、プライバシーも守りやすくなります。その代わり、計算資源が限られるため軽量化が必要です。

## 蒸留

大きなモデルの出力を教師として使い、小さなモデルに同じ振る舞いを学ばせる手法です。正解ラベルだけを使うより、大きなモデルが出す確率の分布から多くの情報を受け取れます。性能を保ったまま小型化できます。

## 宝くじ仮説

大きなネットワークの中には、それだけを取り出して学習しても同等の性能を出せる小さな部分ネットワークが最初から潜んでいる、という仮説です。当たりくじを引くために、たくさん買うようなものだという見立てです。

## プルーニング

重要でない重みや結合を削り取って、モデルを小さくする手法です。枝を刈り込むことにたとえた名前です。削っても精度がほとんど落ちない場合が多いことが知られており、宝くじ仮説とも深く関わっています。

## モデル圧縮

モデルの大きさや計算量を減らす技術の総称です。蒸留、プルーニング、量子化などが含まれます。エッジAIのように、限られた計算資源しかない機器の上で動かすためには欠かせない工夫になります。

## 量子化

重みや計算を、より少ないビット数で表すようにする手法です。たとえば32ビットの小数を8ビットの整数に置き換えます。必要なメモリと計算量が減り、動作が速くなります。精度の落ち方との釣り合いが要点です。

### モデル圧縮＝限られた機器で動かすための工夫

#### 蒸留

大きなモデルを教師に  
小さなモデルへ写す

#### プルーニング

重要でない結合を削る  
宝くじ仮説と関わる

#### 量子化

少ないビット数で表す  
32ビット→8ビットなど

### 手元の機器で動かすのがエッジAI

通信が要らず、応答も速く、データを外へ出さずに済む

図 量子化

## 確認問題 第6部のふりかえり

### 確認問題 第6部

**第1問** 三人の人物が写った画像にセマンティックセグメンテーションを適用した結果として、最も適切な選択肢を1つ選べ

答

**三人がまとめて人というカテゴリの同じ色で塗られる**

セマンティックセグメンテーションは画素ごとにカテゴリを判定しますが、同じカテゴリの個体は区別しません。個体まで分けるのがインスタンスセグメンテーション、両者を統合したものがパノプティックセグメンテーションです。

### 確認問題 第6部

**第2問** ILSVRCで2015年に優勝し、層を飛び越えて入力をそのまま足す構造を導入したモデルとして、最も適切な選択肢を1つ選べ

答

**ResNet**

ResNetはスキップ結合によって、層を深くしても勾配が伝わりやすい構造を作りました。これにより100層を超える深さでも学習できるようになりました。GoogLeNetは前年の2014年の優勝モデルです。

### 確認問題 第6部

**第3問** SENetとDenseNetの構造上の工夫として、最も適切な選択肢を1つ選べ

答

**SENetはチャンネルごとの重要度を学習して重み付けし、DenseNetは各層の出力をそれ以降のすべての層へつなぐ**

SENetは特徴マップのチャンネルごとに重要度を計算して掛け直す仕組みを持ちます。DenseNetは前の層すべての出力を後ろの層へつなぐため、特徴が再利用され、少ないパラメータで高い精度が出ます。

## 131語チェックリスト

自分の言葉で説明できたら✓を入れてください。読めるだけでは足りません。

- |                                       |                                                            |                                                           |
|---------------------------------------|------------------------------------------------------------|-----------------------------------------------------------|
| <input type="checkbox"/> AlexNet      | <input type="checkbox"/> FCN (Fully Convolutional Network) | <input type="checkbox"/> NAS (Neural Architecture Search) |
| <input type="checkbox"/> DeepLab      | <input type="checkbox"/> FPN                               | <input type="checkbox"/> Open Pose                        |
| <input type="checkbox"/> DenseNet     | <input type="checkbox"/> GoogLeNet                         | <input type="checkbox"/> PSPNet                           |
| <input type="checkbox"/> EfficientNet | <input type="checkbox"/> Mask R-CNN                        | <input type="checkbox"/> ResNet                           |
| <input type="checkbox"/> Fast R-CNN   | <input type="checkbox"/> MnasNet                           | <input type="checkbox"/> SegNet                           |
| <input type="checkbox"/> Faster R-CNN | <input type="checkbox"/> MobileNet                         | <input type="checkbox"/> SENet                            |

- |                                             |                                              |                                                    |
|---------------------------------------------|----------------------------------------------|----------------------------------------------------|
| <input type="checkbox"/> SSD                | <input type="checkbox"/> 統計的機械翻訳             | <input type="checkbox"/> 連続値制御                     |
| <input type="checkbox"/> U-Net              | <input type="checkbox"/> A-D変換               | <input type="checkbox"/> CycleGAN                  |
| <input type="checkbox"/> VGG                | <input type="checkbox"/> WaveNet             | <input type="checkbox"/> DCGAN                     |
| <input type="checkbox"/> Vision Transformer | <input type="checkbox"/> 音韻                  | <input type="checkbox"/> Diffusion Model           |
| <input type="checkbox"/> Wide ResNet        | <input type="checkbox"/> 音声合成                | <input type="checkbox"/> NeRF                      |
| <input type="checkbox"/> YOLO               | <input type="checkbox"/> 音声認識                | <input type="checkbox"/> Pix2Pix                   |
| <input type="checkbox"/> 一般物体認識             | <input type="checkbox"/> 音素                  | <input type="checkbox"/> 音声生成                      |
| <input type="checkbox"/> インスタンスセグメンテーション    | <input type="checkbox"/> 隠れマルコフモデル           | <input type="checkbox"/> 画像生成                      |
| <input type="checkbox"/> 姿勢推定               | <input type="checkbox"/> 感情分析                | <input type="checkbox"/> 敵対的生成ネットワーク(GAN)          |
| <input type="checkbox"/> セマンティックセグメンテーション   | <input type="checkbox"/> 高速フーリエ変換(FFT)       | <input type="checkbox"/> 文章生成                      |
| <input type="checkbox"/> 物体検出               | <input type="checkbox"/> スペクトル包絡             | <input type="checkbox"/> Few-shot                  |
| <input type="checkbox"/> 物体識別               | <input type="checkbox"/> パルス符号変調器(PCM)       | <input type="checkbox"/> One-shot                  |
| <input type="checkbox"/> パノプティックセグメンテーション   | <input type="checkbox"/> フォルマント              | <input type="checkbox"/> 自己教師あり学習                  |
| <input type="checkbox"/> BERT               | <input type="checkbox"/> フォルマント周波数           | <input type="checkbox"/> 事前学習                      |
| <input type="checkbox"/> BoW (Bag-of-Words) | <input type="checkbox"/> メル周波数ケプストラム係数(MFCC) | <input type="checkbox"/> 事前学習済みモデル                 |
| <input type="checkbox"/> CBOW               | <input type="checkbox"/> メル尺度                | <input type="checkbox"/> 破滅的忘却                     |
| <input type="checkbox"/> CEC                | <input type="checkbox"/> 話者識別                | <input type="checkbox"/> 半教師あり学習                   |
| <input type="checkbox"/> ELMo               | <input type="checkbox"/> CTC                 | <input type="checkbox"/> CLIP                      |
| <input type="checkbox"/> fastText           | <input type="checkbox"/> A3C                 | <input type="checkbox"/> DALL-E                    |
| <input type="checkbox"/> GLUE               | <input type="checkbox"/> Agent57             | <input type="checkbox"/> Flamingo                  |
| <input type="checkbox"/> n-gram             | <input type="checkbox"/> APE-X               | <input type="checkbox"/> Image Captioning          |
| <input type="checkbox"/> PaLM               | <input type="checkbox"/> DQN                 | <input type="checkbox"/> Text-To-Image             |
| <input type="checkbox"/> Seq2Seq            | <input type="checkbox"/> OpenAI Five         | <input type="checkbox"/> Visual Question Answering |
| <input type="checkbox"/> TF-IDF             | <input type="checkbox"/> PPO                 | <input type="checkbox"/> Unified-IO                |
| <input type="checkbox"/> word2vec           | <input type="checkbox"/> Rainbow             | <input type="checkbox"/> Zero-shot                 |
| <input type="checkbox"/> 感情分析               | <input type="checkbox"/> RLHF                | <input type="checkbox"/> 基盤モデル                     |
| <input type="checkbox"/> 機械翻訳               | <input type="checkbox"/> sim2real            | <input type="checkbox"/> マルチタスク学習                  |
| <input type="checkbox"/> 形態素解析              | <input type="checkbox"/> アルファスター(AlphaStar)  | <input type="checkbox"/> CAM                       |
| <input type="checkbox"/> 構文解析               | <input type="checkbox"/> オフライン強化学習           | <input type="checkbox"/> Grad-CAM                  |
| <input type="checkbox"/> 質問応答               | <input type="checkbox"/> 残差強化学習              | <input type="checkbox"/> LIME                      |
| <input type="checkbox"/> 情報検索               | <input type="checkbox"/> 状態表現学習              | <input type="checkbox"/> Permutation Importance    |
| <input type="checkbox"/> スキップグラム            | <input type="checkbox"/> ダブルDQN              | <input type="checkbox"/> SHAP                      |
| <input type="checkbox"/> 単語埋め込み             | <input type="checkbox"/> デュエリングネットワーク        | <input type="checkbox"/> 説明可能AI (XAI)              |
| <input type="checkbox"/> 分散表現               | <input type="checkbox"/> ドメインランダムマイゼーション     | <input type="checkbox"/> エッジAI                     |
| <input type="checkbox"/> 文書要約               | <input type="checkbox"/> ノイジーネットワーク          | <input type="checkbox"/> 蒸留                        |
| <input type="checkbox"/> ワンホットベクトル          | <input type="checkbox"/> 報酬成形                | <input type="checkbox"/> 宝くじ仮説                     |
| <input type="checkbox"/> 大規模言語モデル(LLM)      | <input type="checkbox"/> マルチエージェント強化学習(MARL) | <input type="checkbox"/> プルーニング                    |
|                                             |                                              | <input type="checkbox"/> モデル圧縮                     |
|                                             |                                              | <input type="checkbox"/> 量子化                       |

## 第7部

# AIの社会実装に向けて

技術分野／2の中項目・23語

- 35 AIプロジェクトの進め方（19語）
- 36 データの収集・加工・分析・学習（4語）

## 35 AIプロジェクトの進め方

→ **この節の目標。** AIプロジェクトをどのように進めるか、全体像と各フェーズで注意すべき点など／AIプロジェクトを進めるにあたって、考えるべき論点や基本となる概念、国内外の議論、並びに事例／サービスやプロダクトとしてAIシステムを世に出す局面で注意すべきこと／AIの運用に必要となるモデルのヘルスマニタリングやライフサイクル管理等／サービスやプロダクトとしてAIシステムを世に出す局面で注意すべきこと／AIの運用にあたって、どのようなことが必要か。

### AIのビジネス活用

AIを事業の価値に結びつけることです。技術としての精度の高さより、どの業務のどこに効かせるかという見極めの方が、成果を左右します。使われない仕組みを作らないために、現場の課題から出発する必要があります。

### AIプロジェクトの進め方

目的をはっきりさせ、データを集めて整え、小さく試して見極め、本番へ移し、運用しながら改善する、という流れで進めます。作って終わりではなく、運用してからが本番である点が、通常の開発と大きく違います。

### BPR

業務そのものを根本から作り直すことです。既存の仕事の流れにAIを足すだけでは、部分的な効率化にとどまり効果が出ないことが多くあります。仕事の進め方自体を見直して初めて、大きな成果につながります。

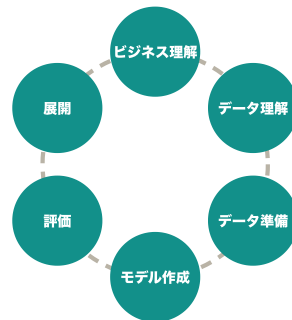
### CRISP-DM

データ分析の進め方をまとめた標準的な手順です。ビジネス理解、データ理解、データ準備、モデル作成、評価、展開の六段階からなります。一方向に進むのではなく、前の段階へ何度も戻りながら進める点が特徴です。

### CRISP-ML

CRISP-DMを、機械学習の実運用に合わせて広げたものです。作って終わりにせず、運用と保守の段階を明確に加えている点が違いです。性能が時間とともに劣化することを前提に置いた手順になっています。

## CRISP-DM=データ分析の標準的な手順



**一方向に進むのではない**

前の段階へ何度も戻りながら進む

=やってみないと分からないから

運用まで含めたものがCRISP-ML

図 CRISP-ML

## Docker

実行に必要な環境をひとまとめにして持ち運べるようにする技術です。自分の手元では動いたのに別の場所では動かない、という問題を防げます。学習した結果を確実に再現するうえでも役立ちます。

## Jupyter Notebook

プログラムと、その実行結果、説明の文章を一つの文書にまとめて書ける環境です。試しながら進める分析の作業に向いており、考えた過程をそのまま記録として残せます。データ分析の定番の道具です。

## MLOps

機械学習のモデルを、作るだけでなく継続して運用していくための考え方と仕組みです。学習、配備、監視、そして再学習という流れを回します。データの傾向が変われば精度は落ちるため、この仕組みが欠かせません。

## PoC

概念実証と訳されます。本格的に作り始める前に、そもそも実現できるのかを小さく試して確かめることです。ここで見極めずに進むと、後から大きな手戻りが生じます。AIは結果が事前に読めないため特に重要です。

## Python

機械学習で最もよく使われているプログラミング言語です。文法が読みやすく、数値計算や機械学習のライブラリが豊富にそろっていることが理由です。データ分析の共通語のような位置づけになっています。

## Web API

作った機能を、ネットワーク越しに呼び出せる形で提供する仕組みです。学習したモデルをこの形で公開すれば、他のシステムから簡単に使えます。AIを既存の業務システムに組み込む際の、標準的なやり方です。

## アジャイル

短い期間で作っては試すことを繰り返しながら進める開発の進め方です。AIは、やってみないとどれだけの精度が出るか分からないため、この進め方と相性がよいとされます。途中で方針を変えやすい点も利点です。

### ウォーターフォール



仕様が固まっている場合に向く

**AIは結果が事前を読めないで、アジャイルの方が合う**

### アジャイル



短く作って試すことを繰り返す

図 アジャイル

## ウォーターフォール

要件定義から設計、実装、試験へと、上流から下流へ順に進める開発の進め方です。仕様が最初から固まっている場合には向きますが、結果が読めないAIの開発には合いにくいとされます。アジャイルと対比して押さえます。

## オープン・イノベーション

自社の中だけで完結させず、外部の知見や技術を取り入れて新しい価値を生み出す考え方です。AIの分野は進歩が速く、必要な技術や人材を自社だけでそろえるのが難しいため、この考え方が重視されます。

## クラウド

計算資源をネットワーク越しに借りて使う仕組みです。高価なGPUを自前で持たなくても、必要なときだけ使えます。データを社外に置くことになるため、機密情報の扱いには別途の判断が必要です。

## 産学連携

企業と、大学や研究機関が協力して研究開発を進めることです。最先端の知見を持つ研究側と、実際の課題とデータを持つ企業側が、互いの足りない部分を補い合います。人材の育成にもつながります。

## ステークホルダーのニーズ

利害関係者が何を求めているかを把握することです。経営層、現場の担当者、そして利用者では、望むものが異なります。技術として優れていても、現場が使わない仕組みは価値を生まないため、初期の把握が欠かせません。

## データサイエンティスト

データから価値を引き出すことを担う専門職です。統計や数理の力、情報技術の力、そして課題を見極めるビジネスの力という三つが求められるとされます。どれか一つだけでは成果につながりません。

## 他企業や他業種との連携

自社だけではデータや知見が足りない場合に、他の企業や別の業種と組むことです。特にデータは、一社で集められる量に限りがあります。連携にあたっては、データの権利や責任の切り分けを契約で定める必要があります。

## 36 データの収集・加工・分析・学習

⇒ **この節の目標。** AIの学習対象となるデータを取得・利用するときに注意すべきこと／集めたデータを加工・分析・学習させるときの注意点／データを共有しながら共同開発を進める場合の留意点。

### アノテーション

データに正解のラベルを付けていく作業です。教師あり学習には欠かせませんが、大量に行うには手間も費用もかかります。付け方の基準がぶれると学習結果も揺らぐため、品質の管理が精度を大きく左右します。

### オープンデータセット

公開されていて誰でも利用できる、データの集まりです。ImageNetなどが代表例で、研究の比較を可能にしました。ただし利用条件はデータごとに異なるため、商用利用の可否などを必ず確認する必要があります。

### コーパス

自然言語処理で使うために集められた、大量の文章のデータです。単に集めただけでなく、品詞や構文の情報が付けられているものもあります。言語モデルの学習は、このコーパスの質と量に大きく左右されます。

### データリーケージ

本来なら使えないはずの情報が学習データに紛れ込み、見かけ上の精度が実際より高く出てしまうことです。検証のときは優秀に見えるのに、運用に移すと全く性能が出ません。気づきにくく、非常に危険な失敗です。

## 確認問題 第7部のふりかえり

### 確認問題 第7部

#### 第1問 CRISP-DMが示す進め方の特徴として、最も適切な選択肢を1つ選べ

答

**業務の理解から始まり、評価の結果に応じて前の工程へ戻る循環的な流れである**

CRISP-DMはデータ分析の標準的な進め方で、業務の理解、データの理解、データの準備、モデル作成、評価、展開という工程を循環させます。機械学習の運用まで広げたものがCRISP-MLです。

### 確認問題 第7部

#### 第2問 AIプロジェクトの開発手法についての記述として、最も不適切な選択肢を1つ選べ

答

**ウォーターフォールは、精度が事前に読みにくいAI開発に最も適した手法である**

AI開発は、やってみるまで精度が分からないという性質があります。工程を戻らない前提のウォーターフォールとは相性が悪く、短い周期で試して見直すアジャイルの考え方が向くとされます。

### 確認問題 第7部

#### 第3問 BPRの説明として、最も適切な選択肢を1つ選べ

答

**業務のやり方そのものを根本から見直して組み立て直すこと**

BPRは業務プロセスの再構築を意味します。AIを導入しても、従来の業務の流れをそのまま残したままでは効果が限られるため、業務の設計自体を見直すことが成果につながります。

## 23語チェックリスト

自分の言葉で説明できたら✓を入れてください。読めるだけでは足りません。

- |                                           |                                       |                                       |
|-------------------------------------------|---------------------------------------|---------------------------------------|
| <input type="checkbox"/> AIのビジネス活用        | <input type="checkbox"/> PoC          | <input type="checkbox"/> ステークホルダーのニーズ |
| <input type="checkbox"/> AIプロジェクトの進め方     | <input type="checkbox"/> Python       | <input type="checkbox"/> データサイエンティスト  |
| <input type="checkbox"/> BPR              | <input type="checkbox"/> Web API      | <input type="checkbox"/> 他企業や他業種との連携  |
| <input type="checkbox"/> CRISP-DM         | <input type="checkbox"/> アジャイル        | <input type="checkbox"/> アノテーション      |
| <input type="checkbox"/> CRISP-ML         | <input type="checkbox"/> ウォーターフォール    | <input type="checkbox"/> オープンデータセット   |
| <input type="checkbox"/> Docker           | <input type="checkbox"/> オープン・イノベーション | <input type="checkbox"/> コーパス         |
| <input type="checkbox"/> Jupyter Notebook | <input type="checkbox"/> クラウド         | <input type="checkbox"/> データリーケージ     |
| <input type="checkbox"/> MLOps            | <input type="checkbox"/> 産学連携         |                                       |

## 第8部

# AIに必要な数理・統計知識

---

技術分野／1の中項目・29語

- 37 AIに必要な数理・統計知識 (29語)

## 37 AIに必要な数理・統計知識

→ **この節の目標。**最適化に必要な数学基礎知識や微分／機械学習で必要となる統計学基礎／基本的な統計情報を計算できる。

### 移動平均

一定の区間をずらしながら平均を取っていく方法です。細かい上下の変動がならされ、全体の傾向が見えやすくなります。株価や気温のような時系列データで、大きな流れをつかむときに使われます。

### 確率分布

確率変数が、どの値をどれくらいの確率で取るかを表したものです。正規分布や二項分布といった代表的な形があります。データがどんな分布に従うかを見立てることが、統計的な分析の出発点になります。

### 確率変数

取る値が確率的に決まる変数のことです。さいころの目や、明日の気温などが当たります。値が一つに定まっている通常の変数とは違い、どの値をどれだけの確からしきで取るかという形で扱います。

### 確率密度

連続した値を取る確率変数について、ある値の周りにどれだけ確率が詰まっているかを表したものです。連続する値は、ちょうどその一点を取る確率が0になってしまいます。そのため確率は、面積として読む形になります。

### 疑似相関

直接の因果関係はないのに、共通の原因があるために関係があるように見えてしまうことです。アイスの売上と水難事故の件数は、どちらも気温という共通の原因によって動きます。相関と因果を混同しない、という戒めです。

### 期待値

確率変数が平均的にどれくらいの値になるかを表したものです。取りうる値と、その確率を掛け合わせて足し上げて求めます。実際に何度も試したときに、その平均が近づいていく値だと考えられます。

### 帰無仮説

統計的検定において、否定したい方の仮説です。多くは差がない、効果がない、と置きます。この仮説のもとでは起こりにくいことが観測された、という筋道で、対立仮説の方を支持するという考え方を取ります。

## 共分散

二つの変数が、どれくらい一緒に動くかを表す量です。一方が大きいときにもう一方も大きければ正、逆向きなら負になります。ただし単位の影響を受けるため、大きさそのものを比べることはできません。

## コサイン類似度

二つのベクトルの向きがどれくらい近いかを測る指標です。マイナス1から1までの値を取り、1に近いほど同じ向きを指します。ベクトルの長さに影響されないため、文書の類似度を測る場面で広く使われます。

## 最小二乗法

予測と実際の差を二乗して足し上げたものが、最も小さくなるようにパラメータを決める方法です。線形回帰で直線を引く際の、最も基本的なやり方です。差を二乗するため、大きな外れが強く効きます。

## 最頻値

データの中で、最も多く現れる値のことです。平均、中央値と並ぶ代表値の一つです。数値でないデータにも使える点が利点である一方、山が二つある分布では一つに定まらないという性質があります。

## 最尤法

実際に観測されたデータが、最も起こりやすくなるようにパラメータを決める方法です。もっともらしさを最大にする、という考え方から名前が付いています。統計モデルを当てはめる際の、標準的な方法です。

## 条件付き確率

ある事象が起きたという条件のもとで、別の事象が起きる確率です。何も分かっていないときの確率とは、値が変わります。ベイズの定理の土台であり、機械学習の確率的な考え方の中心にあります。

## 正規分布

平均を中心として左右対称の、釣鐘のような形をした分布です。ガウス分布とも呼ばれます。身長分布や測定誤差など、自然界や社会の多くの現象がこの形に近づくことが知られており、統計の土台になっています。

## 相関係数

二つの変数の間にある直線的な関係の強さを、マイナス1から1までの値で表したものです。共分散を、それぞれの標準偏差で割って求めます。単位に左右されないため、異なる指標どうしても比べられます。

## 相互情報量

二つの変数が、どれだけ情報を共有しているかを表す量です。片方を知ることによって、もう片方の不確かさがどれだけ減るかを測ります。直線的でない関係も捉えられる点が、相関係数との大きな違いです。

## 対立仮説

帰無仮説が否定されたときに、代わりに採用される仮説です。多くは差がある、効果がある、と置きます。本当に主張したいのはこちらであり、帰無仮説はそれを示すために立てる、という関係になっています。

## 中央値

データを小さい順に並べたとき、ちょうど真ん中に来る値です。極端に大きな値や小さな値があっても影響を受けにくいので、平均より実感に近い代表値になることがあります。所得の分布などで用いられます。

同じデータでも、代表値によって場所が変わる

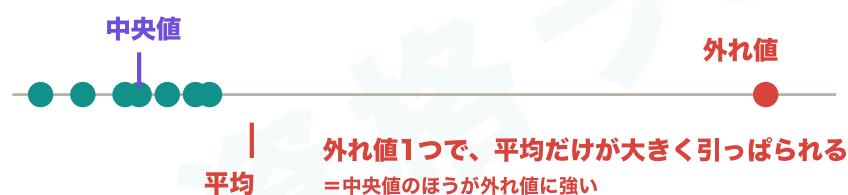


図 中央値

## 度数分布

データを値の範囲ごとに区切り、それぞれに何個入るかを数えて並べたものです。表にしたものが度数分布表、棒で表したものがヒストグラムです。分布の形や偏り、外れ値の有無を目で確かめられます。

## 二項分布

成功する確率が一定の試行をn回繰り返したときに、成功した回数が従う分布です。コインをn回投げて表が出る回数などが当たります。試行が一回だけの場合が、ベルヌーイ分布にあたり、nを大きくすると正規分布に近づきます。

## 外れ値

他のデータから大きく離れた値のことです。測定の誤りである場合もあれば、本当に珍しい現象を捉えている場合もあります。平均や分散を大きく動かすため、除くのか残すのかの判断が分析の結果を左右します。

## 標準偏差

分散の平方根を取ったものです。分散は二乗されているため単位が元と変わってしまいますが、平方根を取ることで元のデータと同じ単位に戻ります。そのため、散らばりの大きさを直感的に読み取れます。

### 正規分布（ガウス分布）

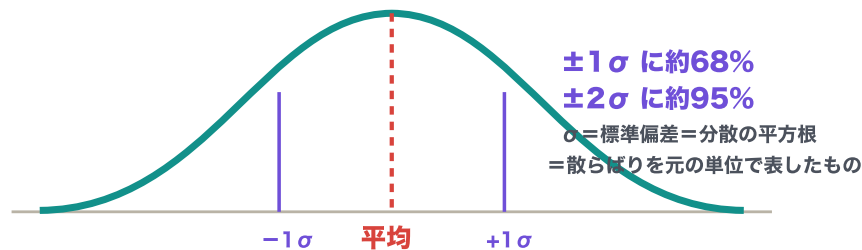


図 標準偏差

## 平均

すべての値を合計して、個数で割ったものです。最もよく使われる代表値ですが、極端に大きな値が一つあるだけで大きく引きずられます。そのため、外れ値のあるデータでは中央値と併せて見る必要があります。

## 分散

それぞれの値が平均からどれだけ離れているかを二乗し、その平均を取ったものです。データの散らばりの大きさを表します。二乗しているため単位が元と変わる点が、標準偏差との違いになります。

## 偏相関係数

第三の変数の影響を取り除いたうえで求めた、二つの変数の相関です。共通の原因によって生じる見かけ上の関係、つまり疑似相関を見抜くために使われます。相関と因果を切り分ける道具の一つです。

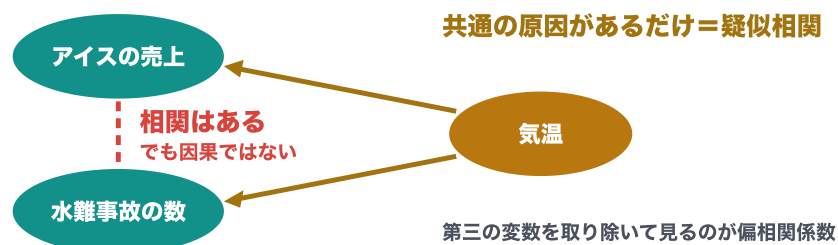


図 偏相関係数

## ベルヌーイ分布

成功か失敗かという、二つの結果しか取らない試行を一回だけ行ったときの分布です。コインを一回投げた場合が当たります。これを $n$ 回繰り返したときの成功回数の分布が、二項分布になるという関係です。

## ポアソン分布

一定の時間や範囲の中で、まれにしか起きない事象が何回起きるかを表す分布です。一時間あたりの来店客数や、事故の発生件数などが当たります。平均と分散が等しくなるという、分かりやすい性質を持ちます。

## マハラノビス距離

データの散らばり方と、変数どうしの相関を考慮して測る距離です。単純な直線距離では近く見えても、分布から見れば大きく外れている点を捉えられます。そのため、外れ値の検出に用いられます。

## ユークリッド距離

二点を結ぶ直線の長さで測る、最も基本的な距離です。三平方の定理をそのまま多次元に広げた形になります。分かりやすい一方、次元が高くなると差がつきにくくなる、次元の呪いの影響を受けます。

## 確認問題 第8部のふりかえり

### 確認問題 第8部

#### 第1問 度数分布を確認することで得られることとして、最も適切な選択肢を1つ選べ

答

**値の散らばり方や、極端に離れた値の有無を目で確かめられる**

度数分布は値を区間ごとに数えて分布の形を示すものです。山が一つか二つか、極端に離れた外れ値があるかといったことは、代表値だけを見ても分からず、分布を見て初めて分かります。

### 確認問題 第8部

#### 第2問 連続的な値をとる確率変数についての記述として、最も適切な選択肢を1つ選べ

答

**ある一点をとる確率は0であり、区間に入る確率は確率密度を積分して求める**

連続的な確率変数では、ちょうどある一点になる確率は0です。意味を持つのは区間に入る確率で、これは確率密度をその区間で積分して求めます。確率密度そのものは確率ではないため、1を超えることもあります。

### 確認問題 第8部

#### 第3問 以下の文章を読み、空欄に最もよく当てはまる選択肢を1つ選べ 出た目に応じて賞金が決まるくじについて、賞金の平均的な大きさを表す値を（ ）という。

答

**期待値**

期待値は、とりうる値にその確率を掛けて足し合わせた値で、長く繰り返したときの平均に相当します。賭けや保険の設計など、不確実な結果の価値を比べるときの基本になります。

## 29語チェックリスト

自分の言葉で説明できたら✓を入れてください。読めるだけでは足りません。

- |                                  |                                 |                                  |
|----------------------------------|---------------------------------|----------------------------------|
| <input type="checkbox"/> 移動平均    | <input type="checkbox"/> 最小二乗法  | <input type="checkbox"/> 度数分布    |
| <input type="checkbox"/> 確率分布    | <input type="checkbox"/> 最頻値    | <input type="checkbox"/> 二項分布    |
| <input type="checkbox"/> 確率変数    | <input type="checkbox"/> 最尤法    | <input type="checkbox"/> 外れ値     |
| <input type="checkbox"/> 確率密度    | <input type="checkbox"/> 条件付き確率 | <input type="checkbox"/> 標準偏差    |
| <input type="checkbox"/> 疑似相関    | <input type="checkbox"/> 正規分布   | <input type="checkbox"/> 平均      |
| <input type="checkbox"/> 期待値     | <input type="checkbox"/> 相関係数   | <input type="checkbox"/> 分散      |
| <input type="checkbox"/> 帰無仮説    | <input type="checkbox"/> 相互情報量  | <input type="checkbox"/> 偏相関係数   |
| <input type="checkbox"/> 共分散     | <input type="checkbox"/> 対立仮説   | <input type="checkbox"/> ベルヌーイ分布 |
| <input type="checkbox"/> コサイン類似度 | <input type="checkbox"/> 中央値    | <input type="checkbox"/> ポアソン分布  |

☐ マハラノビス距離☐ ユークリッド距離

AI資格ラボ

## 第9部

# AIに関する法律と契約

法律・倫理分野／7の中項目・39語

- 1 個人情報保護法（11語）
- 2 著作権法（6語）
- 3 特許法（7語）
- 4 不正競争防止法（2語）
- 5 独占禁止法（2語）
- 6 AI開発委託契約（7語）
- 7 AIサービス提供契約（4語）

## 1 個人情報保護法

→ **この節の目標。**個人情報保護法が適用される場面を理解できている／個人情報等の定義に従い、基本的な事例について個人情報かのあてはめができる／個人情報等に関して生じる常識的な義務について理解しており、具体的な場面において判断ができる／匿名加工情報、仮名加工情報の制度趣旨や最基本事項について理解している／GDPRが適用される場合の概要を理解している。

### GDPR

EUの一般データ保護規則です。EU域内にいる人の個人データの扱いを厳しく定めています。EUに拠点がなくても、域内の人のデータを扱えば適用される点と、違反時の制裁金が非常に高額である点が特徴です。

### 仮名加工情報

他の情報と照らし合わせない限り、特定の個人が分からないように加工した情報です。組織の内部で分析に使うことを想定しており、匿名加工情報と違って、第三者へ提供することは原則としてできません。

### 個人識別符号

それ単体で特定の個人を識別できる符号のことです。指紋や顔の特徴を変換したデータ、旅券番号、マイナンバーなどが当たります。これに該当すれば、氏名が含まれていなくても、それだけで個人情報として扱われます。

### 個人データ

個人情報データベースなどを構成している個人情報のことです。検索できるように体系的にまとめられている点が要点で、単なる個人情報より義務が重くなります。第三者提供の制限などは、この個人データにかかります。

### 個人情報

生存する個人に関する情報で、特定の個人を識別できるものです。他の情報と容易に照らし合わせることで識別できる場合も含まれます。亡くなった方の情報は、原則としてこの定義には含まれません。

### 第三者提供

個人データを、本人以外の第三者へ渡すことです。原則として、あらかじめ本人の同意を得る必要があります。誰にいつ渡したかを記録する義務もあり、AIの学習にデータを渡す場面でも問題になります。

### 匿名加工情報

特定の個人を識別できず、しかも元に戻すこともできないように加工した情報です。この状態にすれば、本人の同意なく第三者へ提供できます。ただし加工の基準が定められており、識別を試みることも禁じられています。



図 匿名加工情報

## 保有個人データ

事業者が、開示、訂正、利用停止などを行う権限を持っている個人データです。本人から開示などを求められた場合、これに応じる義務があります。本人の権利行使の対象となる範囲を定めるための区分です。

## 要配慮個人情報

人種、信条、社会的身分、病歴、犯罪の経歴など、取り扱いに特に配慮を要する情報です。不当な差別につながりかねないため、取得の段階から原則として本人の同意が必要で、通常の個人情報より扱いが厳しくなります。

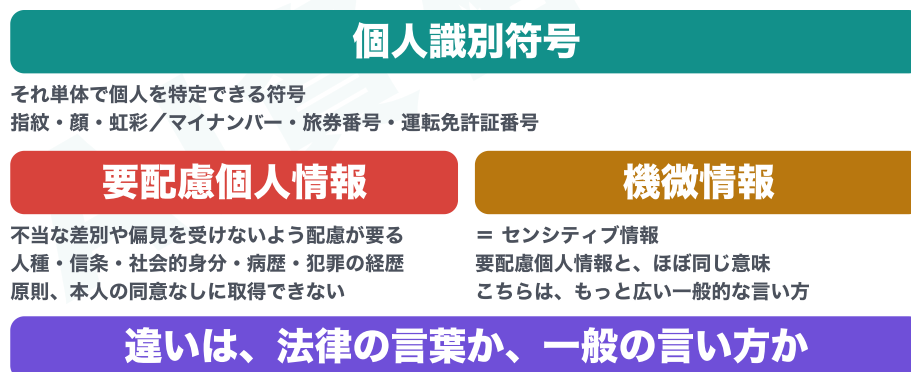


図 要配慮個人情報

## 利用目的

個人情報を何のために使うのかという目的です。取得の際にできる限り特定し、本人に通知するか公表しなければなりません。後から目的を変える場合も、元の目的と関連する範囲に限られるという制限があります。

## 委託

利用目的の達成に必要な範囲で、個人データの取り扱いを外部に任せることです。第三者提供には当たらないため本人の同意は不要ですが、その代わりに委託先を監督する義務を負います。クラウド利用でも論点になります。

## 2 著作権法

→ **この節の目標。** AI開発における生成物の著作権の成否について、具体的な事例であてはめができる／著作権法違反となるコードやデータ等の利用方法をAIの場面に即して理解している／AI生成物に関する著作権について、論点と確立されている考え方について理解している／ライセンスの必要性及びライセンス契約の理解をしている／データの利活用に関する著作権法上の留意点を理解している／著作権法30条4のについて、制度趣旨と精度概要を理解している。

### 創作性

著作物として保護されるための要件で、作者の個性が何らかの形で表れていることを指します。芸術的に優れている必要はありません。一方で、ありふれた表現や、単なる事実やデータそのものには認められません。

### 著作物

思想または感情を創作的に表現したもので、文芸、学術、美術、音楽の範囲に属するものと定義されています。アイデアそのものは保護されず、表現された形が保護の対象になる、という点が重要です。

### AI生成物

AIが生成したもののことです。著作物として保護されるかどうかは、人の創作的な寄与があったかで判断されます。簡単な指示を与えただけでは認められにくく、どこからが創作かという線引きが議論になっています。

### AIが作ったものに著作権はあるか

#### 原則、ありません

著作権は人間の創作的な表現を守るもの  
AIが自動で出しただけでは創作ではない

#### 認められうる

人間が指示や修正を重ねて  
創作的に関わったと言えるなら

#### 著作権侵害

#### 名誉毀損

どちらも「AIが勝手にやった」は通らない。公開した人の責任

図 AI生成物

### 利用規約

サービスを使う際の条件を定めたものです。著作権との関係では、そのサイトの画像や文章を学習に使ってよいかが、ここで定められている場合があります。法律上許されても、契約で禁じられていることがあります。

### 著作権侵害

権利者の許しを得ずに著作物を利用することです。判断は、元の著作物に依拠したかという点と、表現が似ているかという点の両方から行われます。偶然似てしまった場合は、侵害には当たりません。

## 著作権

著作物を独占的に利用できる権利です。著作権法30条の4により、情報解析を目的とする利用は、元の著作物を味わうことを目的としない限り、原則として許されます。これがAIの学習を支える重要な規定です。

## 3 特許法

→ **この節の目標。** 特許権と著作権のすみ分け、異同についてAIの場面に即して基本を理解している／職務発明について基本を理解している／発明、新規性、進歩性について、その趣旨を理解している／特許権と営業秘密との違いについて理解している。

## 発明

自然法則を利用した技術的な思想の創作のうち、高度なものと定義されています。自然法則そのものや、単なる計算方法、人為的な取り決めは発明に当たりません。特許の対象になるかどうかの出発点となる概念です。

## 新規性

まだ世の中に知られていないことという、特許を受けるための要件です。出願の前に論文で発表したり製品として売り出したりすると失われます。うっかり公表してしまうと特許が取れなくなるため、注意が必要です。

## 進歩性

その分野の通常の技術者が、既に知られている技術から容易に思いつくものではないという要件です。新規性があっても、当たり前の組み合わせにすぎない場合は特許になりません。実務で最も争われる要件です。

## 知的財産権

知的な創作活動の成果や、営業上の目印に与えられる権利の総称です。特許権、著作権、商標権、意匠権などが含まれます。権利ごとに、保護の対象、取得の方法、期間が異なる点を整理して押さえます。

### 知的財産権

人が頭で作り出したものを守る権利の総称

①著作権  
表現

登録なしで発生

②特許権  
発明

登録が要る

③商標権  
名前・マーク

登録が要る

④意匠権  
見た目

登録が要る

**登録が要るかが、いちばん問われる**

図 知的財産権

## 発明者

実際に発明を行った自然人のことです。日本を含む多くの国では、AIは発明者になれないという判断が示されています。AIを使って生まれた発明については、誰を発明者とするかが議論になっています。

## 職務発明

従業員が職務として行った発明のことです。企業があらかじめ定めておけば権利を企業のものにできますが、その代わりに従業員へ相当の利益を与える必要があります。企業と発明者の利害を調整する仕組みです。

## 特許権

発明を独占的に実施できる権利です。著作権と違って、出願して審査を通らなければ発生しません。期間は出願の日から二十年で、内容は公開されます。技術を公開する見返りに独占を認める、という考え方にに基づきます。

## 4 不正競争防止法

⇒ **この節の目標。**不正競争防止法による保護の特許権との差異において理解している／営業秘密の三要件の基本事項を理解している／限定提供データの制度趣旨や制度の最基本事項を理解している。

### 営業秘密

不正競争防止法で保護される情報です。秘密として管理されていること、事業に有用であること、公然と知られていないことという三つの要件を、すべて満たす必要があります。一つでも欠けると保護されません。

### 限定提供データ

特定の相手に限って提供される、電磁的に管理された商業的なデータです。営業秘密の三要件を満たさなくても保護の対象になります。データを取引する場面が増えたことを受けて、後から設けられた区分です。

## 不正競争防止法

事業者どうしの、不正な競争を防ぐための法律

### 営業秘密

- ① 秘密として管理されている
  - ② 事業に役立つ情報である
  - ③ 公然と知られていない
- 3つ全部を満たしたもののだけ

### 限定提供データ

特定の相手に限って提供するデータ  
営業秘密ほど厳しく秘密にしない  
でも、誰にでも渡すわけでもない  
その中間を守る区分

**社外の生成AIに貼ると、管理が崩れる**

図 限定提供データ

## 5 独占禁止法

⇒ **この節の目標。** AIと独占禁止法として、どのような事項が論点とされているかについて理解している。

### 競争制限

市場における競争を妨げることです。AIとの関係では、複数の事業者が同じ価格アルゴリズムを使うことで、明示的に相談しなくても価格が同じ方向に動いてしまう、という点が論点になっています。

### 公正競争阻害性

公正な競争を妨げるおそれがあることです。不公正な取引方法に当たるかどうかを判断する際の基準になります。データを大量に持つ事業者が、その優位を使って取引先に不利な条件を課す場面などが問題になります。

## 6 AI開発委託契約

⇒ **この節の目標。** AI開発の各フェーズの内容とそれらのフェーズが置かれている趣旨を理解している／各フェーズの内容と適切な契約関係を理解している／知的財産の帰属と利用条件について理解している／秘密保持契約(NDA)に関する基本的事項について理解している。

### AI・データの利用に関する契約ガイドライン

経済産業省が公表した指針です。AI開発は、やってみないと性能が分からないという特性を持ちます。そこで、いきなり完成を約束するのではなく、段階に分けて進める探索的段階型の開発方式を示しています。

### NDA

秘密保持契約のことです。AI開発では、委託する側がデータを、受託する側が技術やノウハウを開示することになります。どちらの情報も守られるよう、開発の話を始める前の段階で結ぶのが通例です。

### 請負契約

仕事を完成させることを約束する契約です。完成しなければ報酬を受け取れません。AIはデータ次第で結果が変わり、事前に精度を約束しにくいいため、この形は開発の実態に合いにくいとされます。

### 準委任契約

一定の業務を行うことを約束する契約で、完成させる義務までは負いません。AI開発は結果が読めないため、この形が推奨されます。特に、実現できるかを試す段階では、この契約が適しています。



**AIは、やってみないと精度が分からない**  
→ **だから準委任が推奨される（特にPoCの段階）**

図 準委任契約

## 精度保証

一定の精度が出ることを約束することです。AIは学習に使うデータや、実際に入ってくるデータによって性能が変わるため、原則として保証しない形を取ります。その旨を契約に明記しておくことが重要です。

## PoC

概念実証のことです。契約を段階に分ける進め方では、最初の段階として置かれます。ここで実現できる見込みを確かめ、その結果を踏まえて次の開発段階へ進むかどうかを、あらためて判断します。

## 保守契約

納品したあとの運用や保守について定める契約です。AIは時間とともに周囲の状況が変わり精度が落ちるため、いつどのように再学習するのか、その費用は誰が持つのかを、あらかじめ定めておく必要があります。

# 7 AIサービス提供契約

⇒ **この節の目標。**SaaS型における特殊性を理解している／SaaS型の特殊性を踏まえ、保守や知的財産等に関する契約条項について基本事項を理解している。

## SaaS

ソフトウェアを、購入して手元に置くのではなく、ネットワーク越しに使う形で提供する仕組みです。AIの機能も、この形で提供されることが増えています。データが提供側に渡る点が、契約上の論点になります。

## データ利用権

誰がどのデータをどのように使えるかという取り決めです。データには所有権という考え方がそのままは当てはまらないため、契約で定めるほかありません。学習結果を誰のものとするかも、ここで決めます。

## 利用規約

サービスを使う際の条件を定めたものです。AIサービスでは、利用者が入力したデータを学習に使うかどうかを、ここで明示する必要があります。業務で使う際は、この点を必ず確認することが求められます。

## 精度保証

サービスを提供する側が、一定の精度を約束することです。AIの出力は入力によって変わるため、提供側も原則として保証しない旨を利用規約に明記します。利用者側も、出力をそのまま信じない前提が必要です。

AI資格ラボ

## 確認問題 第9部のふりかえり

### 確認問題 第9部

#### 第1問 要配慮個人情報の取り扱いとして、最も適切な選択肢を1つ選べ

答

**原則として、取得にあたって本人の同意が必要であり、本人の同意なく取得してはならない**

要配慮個人情報は、人種、信条、社会的身分、病歴、犯罪の経歴などの、不当な差別や偏見が生じるおそれのある情報です。取得には原則として本人の同意が必要で、本人の同意なく第三者へ提供する仕組みも認められていません。

### 確認問題 第9部

#### 第2問 個人データと保有個人データについての記述として、最も適切な選択肢を1つ選べ

答

**個人データは体系的に整理された情報を構成する個人情報で、そのうち開示や訂正などに応じる権限があるものが保有個人データである**

検索できるように体系的に整理された情報を構成する個人情報が個人データです。そのうち、事業者が開示、訂正、利用停止などに応じる権限を持つものが保有個人データで、本人はこれらの請求を行います。

### 確認問題 第9部

#### 第3問 個人情報を取得する際の利用目的の取り扱いとして、最も不適切な選択肢を1つ選べ

答

**いったん公表した利用目的は、事業者の判断でどのようにでも変更できる**

利用目的の変更は、変更前の目的と関連性を有すると合理的に認められる範囲に限られます。事業者の判断でどのようにでも変えられるわけではありません。

## 39語チェックリスト

自分の言葉で説明できたら✓を入れてください。読めるだけでは足りません。

- |                                  |                                |                                  |
|----------------------------------|--------------------------------|----------------------------------|
| <input type="checkbox"/> GDPR    | <input type="checkbox"/> 利用目的  | <input type="checkbox"/> 新規性     |
| <input type="checkbox"/> 仮名加工情報  | <input type="checkbox"/> 委託    | <input type="checkbox"/> 進歩性     |
| <input type="checkbox"/> 個人識別符号  | <input type="checkbox"/> 創作性   | <input type="checkbox"/> 知的財産権   |
| <input type="checkbox"/> 個人データ   | <input type="checkbox"/> 著作物   | <input type="checkbox"/> 発明者     |
| <input type="checkbox"/> 個人情報    | <input type="checkbox"/> AI生成物 | <input type="checkbox"/> 職務発明    |
| <input type="checkbox"/> 第三者提供   | <input type="checkbox"/> 利用規約  | <input type="checkbox"/> 特許権     |
| <input type="checkbox"/> 匿名加工情報  | <input type="checkbox"/> 著作権侵害 | <input type="checkbox"/> 営業秘密    |
| <input type="checkbox"/> 保有個人データ | <input type="checkbox"/> 著作権   | <input type="checkbox"/> 限定提供データ |
| <input type="checkbox"/> 要配慮個人情報 | <input type="checkbox"/> 発明    |                                  |

- |                                                |                                |                                 |
|------------------------------------------------|--------------------------------|---------------------------------|
| <input type="checkbox"/> 競争制限                  | <input type="checkbox"/> 請負契約  | <input type="checkbox"/> SaaS   |
| <input type="checkbox"/> 公正競争阻害性               | <input type="checkbox"/> 準委任契約 | <input type="checkbox"/> データ利用権 |
| <input type="checkbox"/> AI・データの利用に関する契約ガイドライン | <input type="checkbox"/> 精度保証  | <input type="checkbox"/> 利用規約   |
| <input type="checkbox"/> NDA                   | <input type="checkbox"/> PoC   | <input type="checkbox"/> 精度保証   |
|                                                | <input type="checkbox"/> 保守契約  |                                 |

AI資格ラボ

## 第10部

# AI倫理・AIガバナンス

法律・倫理分野／11の中項目・45語

- 8 国内外のガイドライン (6語)
- 9 プライバシー (2語)
- 10 公平性 (6語)
- 11 安全性とセキュリティ (6語)
- 12 悪用 (2語)
- 13 透明性 (3語)
- 14 民主主義 (3語)
- 15 環境保護 (2語)
- 16 労働政策 (3語)
- 17 その他の重要な価値 (4語)
- 18 AIガバナンス (8語)

## 8 国内外のガイドライン

→ **この節の目標。**各ガイドラインを通じ共通で議論されている事項を理解している／ソフトロー・ハードローやリスクベースアプローチなどの重要な概念を理解し、そのメリット・デメリットを理解している。

### AI倫理

AIを作り、使ううえで守るべき価値や考え方のことです。法律で禁じられていなくても、してよいとは限りません。公平性、透明性、人間の尊重といった原則が、国内外の指針として整理されています。

### AIガバナンス

AIを組織としてどう管理し、統制していくかという仕組みのことです。方針を定め、体制を作り、監視して見直す、という流れを回します。個々の技術者の良心に任せるのではなく、組織の仕組みとして担保します。



図 AIガバナンス

### 価値原則

AIを扱ううえで基礎に置くべき、共通の価値のことです。人間の尊厳、多様性と包摂、持続可能な社会といった形で示されます。個別の規則を作る前提となる、上位の考え方として位置づけられます。



図 価値原則

### ハードロー

法律や規則のように、国家によって強制力を持って定められた決まりのことです。違反すれば罰則や制裁があります。確実に守らせられる一方、技術の進歩に対して作るのが遅れがちだという弱点があります。

## ソフトロー

指針や行動規範のように、法的な強制力を持たない決まりのことです。状況の変化に合わせて素早く作り直せるため、進歩の速いAIの分野で重視されます。守るかどうかは、事業者の判断に委ねられます。

## リスクベースアプローチ

AIの用途が持つ危険の大きさに応じて、規制の強さを変えるという考え方です。EUのAI法がこの立場を取っています。すべてを一律に厳しくするのではなく、危険の大きいものに規制を集中させます。



図 リスクベースアプローチ

## 9 プライバシー

⇒ **この節の目標。** プライバシー上の問題の所在と、プライバシーが問題となった著名な事例を理解している／データ収集段階と推論段階でプライバシー上の問題が区別できることを理解している／プライバシー上の問題に対応するための方策を理解している／カメラ画像利活用ガイドブックなどに照らし、カメラ画像を利用するAIにおけるプライバシー上留意すべき事項や対応策などを理解している。

### カメラ画像利活用ガイドブック

カメラで撮影した画像を事業に使う際の、実務上の留意点をまとめた指針です。撮影していることをどう知らせるか、取得したデータをどう扱うかなどを示しています。防犯カメラや店舗での人流分析が対象になります。

### プライバシー・バイ・デザイン

あとから対策を足すのではなく、設計の段階からプライバシーへの配慮を組み込むという考え方です。作ってしまってから守ろうとしても、構造としてどうにもならない部分が残ります。最初から前提として織り込む姿勢を指します。

## 10 公平性

⇒ **この節の目標。** 公平性の問題としてどのような問題が存在するのか理解している／公平性に関する代表的な事例について理解している／公平性の問題が生じる原因について理解している／公平性に対処するための要検討事項を理解している／公平性に対処するための技術の基礎について理解している。

## アルゴリズムバイアス

AIの判断に、特定の属性の人が不利になるような偏りが生じることです。学習データに含まれる社会の偏りが、そのまま、あるいは増幅されて出力に現れます。採用や与信の判断で特に問題になります。

## 公平性の定義

何をもって公平とするかは、一つに定まらないという問題です。属性ごとに合格率をそろえるのか、同じ能力の人が同じ扱いを受けるようにするのかで、答えが変わります。しかも複数の定義は同時には満たしません。

## サンプリングバイアス

データの集め方に偏りがあるために、全体を正しく反映しなくなることです。特定の地域や年代からしか集めていなければ、それ以外の人に対して精度が落ちます。データを集める段階で生じる偏りです。

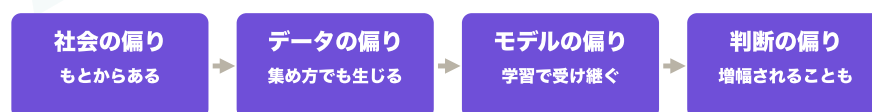
## センシティブ属性

人種、性別、信条、病歴など、差別につながりかねない属性のことです。公平性を考えるうえで、特に注意して扱う必要があります。ただし単に入力から外すだけでは、代理変数を通じて影響が残ります。

## 代理変数

センシティブ属性そのものではないのに、結果としてそれを言い当ててしまう変数のことです。郵便番号が人種と強く結びつく、といった例があります。属性を入力から外しても差別が残る原因になります。

**偏りは、集める段階から出力まで受け継がれる**



**センシティブ属性を入力から外しても、代理変数を通じて残る**  
郵便番号が人種と強く結びつく、といった形で言い当ててしまう

図 代理変数

## データの偏り

学習に使うデータが、現実の分布を正しく映していない状態です。社会に元からある偏りが反映される場合と、集め方の問題で生じる場合があります。どちらも、そのままモデルの偏りとして現れます。

## 11 安全性とセキュリティ

→ **この節の目標。**安全性に関する論点の所在と代表的な事例を理解している／セキュリティ上の課題としてどのような攻撃等が存在しているのか理解している／安全性やセキュリティの課題への対応手段を理解している。

### Adversarial Attack Adversarial Examples

人の目には分からないほど小さな変更を入力に加えることで、AIに誤った判断をさせる攻撃です。標識に小さなシールを貼るだけで別の標識と誤認させる、といった例があります。自動運転などで深刻な脅威になります。

### セキュリティ・バイ・デザイン

設計の段階から安全性を組み込むという考え方です。作ってから対策を追加するのでは、守り切れない部分が残ります。プライバシー・バイ・デザインと同じく、最初から前提として織り込む姿勢を指します。

### データ汚染

学習に使うデータに、悪意のあるデータを混ぜ込む攻撃です。特定の入力にだけ誤った反応をする裏口を仕込むこともできます。外部から集めたデータを学習に使う場合、特に注意が必要になります。

### データ窃取

学習に使われたデータを、モデルの出力から推測して盗み出す攻撃です。ある人のデータが学習に含まれていたかどうかを言い当てられる場合もあり、個人情報の漏えいにつながります。医療データなどでは特に深刻です。

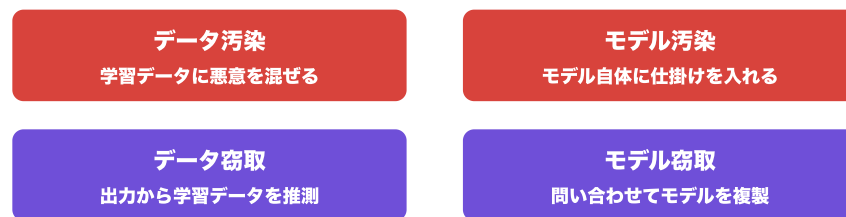
### モデル窃取

公開されているモデルに何度も問い合わせ、その入出力からモデルそのものを再現してしまう攻撃です。多額の費用をかけて作ったモデルが、安価に複製されることになります。知的財産を守るうえでの脅威です。

### モデル汚染

モデルそのものに手を加えて、意図しない振る舞いを仕込む攻撃です。公開されている学習済みモデルを取り込んで使う場合、そこに仕掛けが含まれている危険があります。出どころの確認が欠かせません。

### AIへの攻撃＝「入れる側」と「盗む側」に分かれる



### Adversarial Attack＝人には分からない小さな変更で誤らせる

図 モデル汚染

## 12 悪用

⇒ **この節の目標。** AI技術の悪用の例とその影響の意味を理解している／悪用事例ごとに対応策を理解している。

### ディープフェイク

ディープラーニングを使って作られた、本物と見分けがつかない偽の映像や音声です。有名人に言っていないことを言わせる、といった悪用が起きています。名誉毀損や詐欺、世論の操作につながる危険があります。

### フェイクニュース

事実でない情報を、本当のニュースのように見せかけて広めるものです。生成AIによって、大量に、しかも自然な文章で作れるようになりました。作る側の手間が下がったことが、問題を深刻にしています。

## 13 透明性

⇒ **この節の目標。** 透明性、説明可能性について求められる根拠と内容を理解している／説明可能性や透明性を確保するにあたって考慮すべき事項を理解している／説明可能性確保技術の代表例の概要を理解している／透明性を与える対象について代表的な事項を理解している。

### データの来歴

そのデータがどこから来て、どう加工されてきたかという履歴のことです。学習に使ったデータの出どころをたどれるようにしておくことで、権利や偏りの問題が起きたときに原因を確かめられます。

### 説明可能性

AIがなぜその判断をしたのかを、人が理解できる形で示せることです。医療や与信のように、判断の理由が問われる場面では欠かせません。性能の高さと説明のしやすさは、しばしば両立が難しくなります。

## ブラックボックス

中で何が起きているのかが分からない状態を指します。ディープラーニングは層が深く、判断の理由を人が追えません。精度は高いのに理由が説明できない、という点が実務での採用をためらわせる原因になります。

## 14 民主主義

⇒ **この節の目標。** 民主主義に対してどのような影響があるのかを理解している。

### エコーチェンバー

同じ意見の人ばかりが集まる場で、似た主張が反響し合い、その考えが正しいと強く思い込んでしまう現象です。閉じた小部屋で声が反響する様子にたとえた言葉で、意見の対立を深める一因とされています。

### フィルターバブル

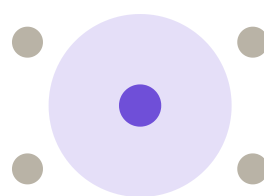
推薦の仕組みが利用者の好みに合う情報ばかりを見せるため、それ以外の情報に触れられなくなる現象です。泡に包まれた状態にたとえられます。本人は偏っていることに気づけない点が、特に問題とされます。

エコーチェンバー



同じ意見の人どうして反響し合う  
どちらも、本人は偏りに気づけない

フィルターバブル



好みに合う情報だけに囲まれる

図 フィルターバブル

### フェイクニュース

民主主義との関係では、偽の情報が選挙や世論に影響を与えることが問題になります。生成AIによって大量に作れるようになったうえ、推薦の仕組みが拡散を後押しするため、影響がより大きくなっています。

## 15 環境保護

⇒ **この節の目標。** 環境保護とAIについてどのような点が議論されているのか理解している。

## 気候変動

AIとの関係では、二つの側面があります。一つは、大規模なモデルの学習が大量の電力を使い、二酸化炭素の排出につながるという負の側面です。もう一つは、気象予測や省エネにAIが役立つという正の側面です。

## モデル学習の電力消費

大規模なモデルの学習には、膨大な計算と、それに伴う大量の電力が必要になります。一度の学習で相当量の二酸化炭素を排出するという試算もあります。モデルの軽量化は、環境の観点からも意味を持ちます。

# 16 労働政策

⇒ **この節の目標。** AIが雇用に与える影響について理解している。

## AIとの協働

AIに人の仕事を置き換えさせるのではなく、人とAIが役割を分け合って働くという考え方です。判断はAIに補助させ、最終的な責任は人が持つ、という形が現実的だとされます。人間の関与とも深く関わります。

## スキルの喪失

AIに任せきりにすることで、人がもともと持っていた技能を失ってしまうことです。異常が起きてAIが使えなくなったとき、人が対処できなくなる危険があります。自動化が進むほど深刻になる問題です。

## 労働力不足

人口の減少によって働き手が足りなくなる問題です。AIによる自動化は、これを補う手段として期待されています。雇用を奪うという懸念と、足りない働き手を補うという期待の、両面から議論されます。

# 17 その他の重要な価値

⇒ **この節の目標。** インクルージョン、軍事利用、自律性などのAIの様々な課題について問題の所在を理解している。

## インクルージョン

多様な人々を排除せず、誰もが使える形にするという考え方です。特定の属性の人だけ精度が落ちるAIは、この考え方に反します。開発する側の多様性が、作られるものの多様性にも影響するとされます。

## 軍事利用

AIを兵器や軍事の目的に使うことです。人の判断を介さずに攻撃を決める自律型致死兵器システムについては、国際的に規制を求める議論が続いています。研究者が開発への協力を拒む動きもあります。

## 死者への敬意

亡くなった方の声や姿を、生成AIで再現することをめぐる問題です。遺族の慰めになるという見方がある一方、本人の同意を得ようがない点が問われます。個人情報保護法は生存する人が対象であるため、法律だけでは扱えません。

## 人間の自律性

人が自分で考え、自分で決める力のことです。AIによる推薦や助言が強くなりすぎると、知らないうちに判断を誘導されてしまいます。本人が気づかないまま行動が変えられることが、特に問題とされます。

# 18 AIガバナンス

⇒ **この節の目標。** AI倫理アセスメントの必要性について理解している／人間やステークホルダー関与について、その意味と必要性を理解している／AI倫理を実現するための組織体制の在り方について理解している／AI倫理上の課題に対処するためのその他の様々な手法について理解している。

## AIポリシー

組織としてAIをどう扱うかを定めた方針のことです。何に使ってよいか、どんな確認を経るか、誰が責任を持つかを明文化します。現場ごとの判断に任せず、組織として一貫した扱いを保つための土台になります。

## ダイバーシティ

多様性のことです。開発に関わる人が偏っていると、気づかれない偏りがそのまま製品に入り込みます。多様な視点を入れることが、公平性を保つための実践的な手立てとして位置づけられています。

## AIに対する監査

AIの開発や運用が、定めた方針や基準に沿っているかを、独立した立場から確かめることです。作った本人だけで確認しても見落としが生じるため、外部や第三者の目を入れることに意味があります。

## 倫理アセスメント

AIを作る前や導入する前に、倫理的な問題が起きないかをあらかじめ評価することです。誰が不利益を受ける可能性があるかを洗い出します。問題が起きてから対応するより、はるかに費用も影響も小さく済みます。

## 人間の関与

AIの判断に、人が関わる仕組みを残すことです。人が最終判断を行う形、人が確認して覆せる形など、関わり方には段階があります。重い結果を招く判断ほど、人の関与を厚くすることが求められます。

## モニタリング

運用を始めたあとも、AIの振る舞いを継続して監視することです。データの傾向が変われば精度は落ち、思わぬ偏りも現れます。作ったときの性能がそのまま続くわけではない、という前提に立った取り組みです。

## 再現性

同じ条件で行えば同じ結果が得られることです。学習には乱数が使われるため、条件を記録しておかないと再現できません。問題が起きたときに原因を確かめるためにも、再現できる状態を保つ必要があります。

## トレーサビリティ

どのデータを使い、どのモデルで、いつ、どんな判断をしたのかを、後からたどれるようにしておくことです。問題が起きた際の原因究明と、説明責任を果たすための土台になります。データの来歴とも深く関わります。

## 確認問題 第10部のふりかえり

### 確認問題 第10部

#### 第1問 ハードローとソフトローの違いとして、最も適切な選択肢を1つ選べ

答

**ハードローは法的な拘束力を持ち、ソフトローは指針など拘束力のない規範を指す**

ハードローは法律や規則のように法的拘束力を持つ規範です。ソフトローはガイドラインや業界の自主規制のように拘束力はないものの、実務上の指針として機能します。技術の変化が速い分野では、まずソフトローで対応することが多くなります。

### 確認問題 第10部

#### 第2問 リスクベースアプローチの考え方として、最も適切な選択肢を1つ選べ

答

**用途ごとのリスクの大きさに応じて、規制や対策の強さを変える**

リスクベースアプローチは、同じ技術でも用途によって影響の大きさが違うことをふまえ、リスクに応じて求める対策の強さを変える考え方です。欧州のAIに関する規則でも採られており、AIガバナンスの基本的な発想になっています。

### 確認問題 第10部

#### 第3問 プライバシー・バイ・デザインの考え方として、最も適切な選択肢を1つ選べ

答

**企画や設計の段階からプライバシーへの配慮を組み込んでおく**

プライバシー・バイ・デザインは、後から対策を足すのではなく、設計の初めから配慮を組み込む考え方です。取得する情報を必要最小限にとどめる、初期設定を保護的にするといった形で具体化されます。

## 45語チェックリスト

自分の言葉で説明できたら✓を入れてください。読めるだけでは足りません。

- |                                         |                                                                    |                                   |
|-----------------------------------------|--------------------------------------------------------------------|-----------------------------------|
| <input type="checkbox"/> AI倫理           | <input type="checkbox"/> 公平性の定義                                    | <input type="checkbox"/> データ窃取    |
| <input type="checkbox"/> AIガバナンス        | <input type="checkbox"/> サンプリングバイアス                                | <input type="checkbox"/> モデル窃取    |
| <input type="checkbox"/> 価値原則           | <input type="checkbox"/> センシティブ属性                                  | <input type="checkbox"/> モデル汚染    |
| <input type="checkbox"/> ハードロー          | <input type="checkbox"/> 代理変数                                      | <input type="checkbox"/> ディープフェイク |
| <input type="checkbox"/> ソフトロー          | <input type="checkbox"/> データの偏り                                    | <input type="checkbox"/> フェイクニュース |
| <input type="checkbox"/> リスクベースアプローチ    | <input type="checkbox"/> Adversarial Attack (Adversarial Examples) | <input type="checkbox"/> データの来歴   |
| <input type="checkbox"/> カメラ画像利活用ガイドブック | <input type="checkbox"/> セキュリティ・バイ・デザイン                            | <input type="checkbox"/> 説明可能性    |
| <input type="checkbox"/> プライバシー・バイ・デザイン | <input type="checkbox"/> データ汚染                                     | <input type="checkbox"/> ブラックボックス |
| <input type="checkbox"/> アルゴリズムバイアス     |                                                                    |                                   |

- |                                     |                                   |                                   |
|-------------------------------------|-----------------------------------|-----------------------------------|
| <input type="checkbox"/> エコーチェンバー   | <input type="checkbox"/> 労働力不足    | <input type="checkbox"/> AIに対する監査 |
| <input type="checkbox"/> フィルターバブル   | <input type="checkbox"/> インクルージョン | <input type="checkbox"/> 倫理アセスメント |
| <input type="checkbox"/> フェイクニュース   | <input type="checkbox"/> 軍事利用     | <input type="checkbox"/> 人間の関与    |
| <input type="checkbox"/> 気候変動       | <input type="checkbox"/> 死者への敬意   | <input type="checkbox"/> モニタリング   |
| <input type="checkbox"/> モデル学習の電力消費 | <input type="checkbox"/> 人間の自律性   | <input type="checkbox"/> 再現性      |
| <input type="checkbox"/> AIとの協働     | <input type="checkbox"/> AIポリシー   | <input type="checkbox"/> トレーサビリティ |
| <input type="checkbox"/> スキルの喪失     | <input type="checkbox"/> ダイバーシティ  |                                   |

AI資格ラボ

**巻末 総索引 全495語****A-D変換 p.76**

アナログの音の波を、コンピュータが扱える数値に変換することです。一定の間隔で値を拾う標本…

**A3C p.78**

複数のエージェントを並列に動かし、それぞれの経験をまとめて学習する手法です。同時に多様な…

**AIとの協働 p.123**

AIに人の仕事を置き換えさせるのではなく、人とAIが役割を分け合って働くという考え方です…

**AIに対する監査 p.124**

AIの開発や運用が、定めた方針や基準に沿っているかを、独立した立場から確かめることです。…

**AIのビジネス活用 p.92**

AIを事業の価値に結びつけることです。技術としての精度の高さより、どの業務のどこに効かせ…

**AIガバナンス p.117**

AIを組織としてどう管理し、統制していくかという仕組みのことです。方針を定め、体制を作り…

**AIプロジェクトの進め方 p.92**

目的をはっきりさせ、データを集めて整え、小さく試して見極め、本番へ移し、運用しながら改善…

**AIポリシー p.124**

組織としてAIをどう扱うかを定めた方針のことです。何に使うてよいか、どんな確認を経るか、…

**AI・データの利用に関する契約ガイドライン p.111**

経済産業省が公表した指針です。AI開発は、やってみないと性能が分からないという特性を持ち…

**AI倫理 p.117**

AIを作り、使ううえで守るべき価値や考え方のことです。法律で禁じられていなくても、してよ…

**AI効果 p.6**

AIによって何かが実現されると、その仕組みが分かった途端に「これは単なる自動化であって知…

**AI生成物 p.108**

AIが生成したもののことです。著作物として保護されるかどうかは、人の創作的な寄与があった…

**AMSBound p.44**

AMSGradに、AdaBoundと同じように学習率の上下の制限を加えた手法です。学習の…

**APE-X p.79**

経験を集める役割と、そこから学習する役割を分けて、大規模に並列化した手法です。多数のエー…

**Actor-Critic p.29**

行動を決める役割のActorと、その良さを見積もる役割のCriticを組み合わせる強化学…

**AdaBound p.44**

Adamの学習率に上限と下限を設ける手法です。学習の初めはAdamのように速く進み、終わ…

**AdaDelta p.44**

AdaGradの、学習が進むほど学習率が下がすぎるという弱点を改善した手法です。過去す…

**AdaGrad p.44**

パラメータごとに学習率を変える手法です。これまでよく更新されてきたパラメータほど学習率を…

**Adam p.44**

モーメンタムと、RMSpropによる学習率の調整を組み合わせた手法です。設定をあまり調整…

**Adversarial Attack (Adversarial Examples) p.126**

人の目には分からないほど小さな変更を入力に加えることで、AIに誤った判断をさせる攻撃です…

**Agent57 p.78**

DeepMindが開発したエージェントで、Atariの57種類のゲームすべてで人間の水準…

**AlexNet p.67**

2012年のILSVRCで優勝した畳み込みニューラルネットワークです。トロント大学のヒン…

**Atrous Convolution p.51**

フィルタの要素の間に隙間を空けて畳み込む方法です。Dilated Convolution…

**Attention p.58**

入力のどこに注目すべきかを、重みとして計算する仕組みです。系列を一つの固定長のベクトルに…

**BERT p.72**

2018年にGoogleが発表した言語モデルです。Transformerのエンコーダを使…

**BPR p.92**

業務そのものを根本から作り直すことです。既存の仕事の流れにAIを足すだけでは、部分的な効…

**BPTT p.56**

時間方向に展開したうえで誤差逆伝播法を行う方法です。回帰結合層は同じ重みを何度も使うため…

**BoW (Bag-of-Words) p.90**

文書を、語がそれぞれ何回出てきたかという回数だけで表す方法です。袋に語を放り込んだ状態に…

**Brightness p.61**

画像全体の明るさを変えて、学習データを水増しする方法です。暗い場所で撮られた写真でも同じ…

**CAM p.86**

畳み込みニューラルネットワークが、画像のどこを見て判断したのかを可視化する手法です。グロ…

**CBOW p.72**

word2vecの学習方式の一つです。周りにある語から、真ん中の語が何かを当てさせること…

**CEC p.72**

LSTMの内部で、情報をそのまま保ち続ける記憶セルのことです。ここを通る経路では値が掛け…

**CLIP p.84**

画像と、それを説明する文章を、同じ空間の中で近い位置に置くように学習したモデルです。Op…

**CPU p.38**

コンピュータの中心となる汎用の計算装置です。条件分岐の多い複雑な処理を、順番に手早くこな…

**CRISP-DM p.92**

データ分析の進め方をまとめた標準的な手順です。ビジネス理解、データ理解、データ準備、モデ…

**CRISP-ML p.92**

CRISP-DMを、機械学習の実運用に合わせて広げたものです。作って終わりにせず、運用と…

**CTC p.78**

入力と出力の長さが違う系列を、どこがどこに対応するかを人が与えなくても学習できるようにす…

**Contrast p.61**

画像の明暗の差を強めたり弱めたりして、学習データを水増しする方法です。撮影したときの条件…

**Contrastive Loss p.40**

二つのデータが同じ仲間かどうかを学ばせるための誤差関数です。同じ仲間なら距離が近くなるよ…

**Crop p.61**

画像の一部を切り抜いて、学習データを水増しする方法です。対象が画面のどこにあっても、また…

**CutMix p.61**

二枚の画像の一部を切り貼りして混ぜ、正解ラベルも面積の比率で混ぜる方法です。Cutout…

**Cutout p.62**

画像の一部を四角く塗りつぶして隠す方法です。一部が隠れていても判断できるようになり、画像…

**CycleGAN p.81**

対になっていない画像どうしても、変換の仕方を学習できるGANです。馬の画像とシマウマの画…

**Cycプロジェクト p.15**

人間が持つ一般常識を、すべてコンピュータに書き込もうという試みです。1984年にダグラス…

**DALL-E p.84**

文章の指示から画像を作り出すモデルです。OpenAIが発表し、テキストから画像を生成する…

**DCGAN p.81**

GANに畳み込みニューラルネットワークを取り入れたモデルです。学習が不安定になりやすいG…

**DENDRAL p.15**

1960年代にスタンフォード大学で作られた、世界初のエキスパートシステムとされるプログラ…

**DQN p.79**

Q学習にディープラーニングを組み合わせた手法です。DeepMindがAtariのゲームで…

**DeepLab p.67**

セマンティックセグメンテーションのためのモデルです。フィルタに隙間を空けるAtrous …

**DenseNet p.67**

各層の出力を、それ以降のすべての層へつなぐ構造を持つネットワークです。前の層で得た特徴を…

**Depthwise Separable Convolution p.51**

畳み込みを、空間方向の処理とチャネル方向の処理の二段階に分ける方法です。まとめて行う通常…

**Diffusion Model p.82**

雑音だらけの状態から、少しずつ雑音を取り除いていくことで画像を作り出すモデルです。学習の…

**Dilated Convolution p.51**

Atrous Convolutionと同じで、フィルタに隙間を空けて広い範囲を見る畳み込…

**Docker p.93**

実行に必要な環境をひとまとめにして持ち運べるようにする技術です。自分の手元では動いたのに…

**ELMo p.72**

文脈に応じて、同じ語でも違うベクトルを与える手法です。双方向のLSTMを使って前後の文脈…

**EfficientNet p.67**

ネットワークの深さ、幅、入力画像の大きさの三つを、決まった比率でまとめて拡大する設計のモ…

**FCN (Fully Convolutional Network) p.89**

全結合層をなくし、畳み込み層だけで構成したネットワークです。出力が二次元のまま保たれるた…

**FPN p.68**

解像度の異なる複数の特徴マップを組み合わせる構造です。浅い層の細かい情報と、深い層の意味…

**Fast R-CNN p.67**

物体検出のモデルで、R-CNNを高速にしたものです。領域ごとに何度も畳み込むのではなく、…

**Faster R-CNN p.67**

Fast R-CNNをさらに速くしたモデルです。どこに物体がありそうかという領域の候補づ…

**Few-shot p.83**

ごく少数の例だけを見せて、新しい課題に対応させることです。大規模言語モデルでは、指示文の…

**Flamingo p.84**

画像と文章が交互に混ざった入力を、そのまま扱える視覚言語モデルです。少数の例を示すだけで…

**GDPR p.106**

EUの一般データ保護規則です。EU域内にいる人の個人データの扱いを厳しく定めています。E…

**GLUE p.72**

自然言語を理解する能力を測るための、複数の課題をまとめたベンチマークです。文の含意関係の…

**GPU p.38**

もとは画像処理のために作られた演算装置です。単純な計算を大量に並列で実行できるため、行列…

**GRU p.56**

LSTMを簡単にしたゲート機構です。LSTMが三つのゲートを持つのに対し、GRUは二つに…

**GoogLeNet p.68**

2014年のILSVRCで優勝したモデルです。大きさの違う複数のフィルタを並列に適用する…

**Grad-CAM p.86**

勾配の情報を使って、判断の根拠となった場所を可視化する手法です。CAMと違い、ネットワー…

**ILSVRC p.18**

ImageNetを使って画像認識の精度を競うコンテストです。2012年にトロント大学のチ…

**Image Captioning p.85**

画像を見て、その内容を説明する文章を作る課題です。画像を理解する力と、文章を組み立てる力…

**ImageNet p.18**

1400万枚を超える画像に、何が写っているかのラベルを付けた大規模なデータセットです。ス…

**Jupyter Notebook p.93**

プログラムと、その実行結果、説明の文章を一つの文書にまとめて書ける環境です。試しながら進…

**L0正則化 p.41**

0ではないパラメータの個数そのものを罰する正則化です。不要なパラメータを本当に消せますが…

**L1正則化 p.41**

パラメータの絶対値の合計を罰として加える正則化です。不要なパラメータの値がちょうど0にな…

**L2正則化 p.41**

パラメータの二乗の合計を罰として加える正則化です。値をちょうど0にはしにくい一方で、全体…

**LIME p.86**

ある予測の周辺だけを、単純で分かりやすいモデルで近似して説明する手法です。中身の分からな…

**LSTM p.56**

長い間隔を空けた依存関係を覚えていられるようにした回帰結合層です。情報をためておく記憶セ…

**LeNet p.18**

1998年にヤン・ルカンが発表した、畳み込みニューラルネットワークです。手書きの数字を認…

**Leaky ReLU関数 p.39**

ReLU関数の、入力が負の側をわずかな傾きに変えたものです。ReLUでは負の入力に対して…

**MLOps p.93**

機械学習のモデルを、作るだけでなく継続して運用していくための考え方と仕組みです。学習、配…

**Mask R-CNN p.68**

Faster R-CNNに、物体ごとの領域を塗り分ける枝を追加したモデルです。物体の位置…

**Mini-Max法 p.13**

二人で交互に指すゲームで先を読むための方法です。自分の手番では自分にとって最も良い手を、…

**Mixup p.62**

二枚の画像を重み付きで重ね合わせ、正解ラベルも同じ比率で混ぜる方法です。中間の入力に対し…

**MnasNet p.68**

ネットワークの構造を自動で探索するNASを使い、さらにスマートフォン上での実際の処理速度…

**MobileNet p.68**

スマートフォンなど、計算資源の限られた機器で動かすために作られた軽量なモデルです。畳み込…

**Multi-Head Attention p.58**

Attentionを複数、並列に行う仕組みです。それぞれが違う観点で関係を捉えるため、語…

**NAS (Neural Architecture Search) p.89**

ネットワークの構造そのものを、人ではなく機械が自動で探索する技術です。層の並べ方やフィル…

**NDA p.111**

秘密保持契約のことです。AI開発では、委託する側がデータを、受託する側が技術やノウハウを…

**NeRF p.82**

いくつかの方向から撮った写真をもとに、その場面を三次元として再現し、撮っていない角度から…

**One-shot p.83**

例をたった一つだけ見せて、新しい課題に対応させることです。Few-shotの中でも、特に…

**Open Pose p.68**

画像から人の姿勢を推定するモデルです。複数の人が写っていても、それぞれの関節の位置と、ど…

**OpenAI Five p.79**

OpenAIが開発した、Dota 2というゲームのエージェントです。五人一組で戦う複雑な…

**PPO p.79**

方策を一度にどれだけ変えてよいかに制限を設けることで、学習を安定させた方策勾配法です。実…

**PSPNet p.69**

セマンティックセグメンテーションのモデルです。複数の大きさにプーリングした結果を組み合わ…

**PaLM p.73**

Googleが開発した大規模言語モデルです。膨大なパラメータを持ち、推論を要する課題でも…

**Permutation Importance p.86**

ある特徴の値だけをでたらめに並べ替え、そのとき精度がどれだけ落ちるかで、その特徴の重要さ…

**Pix2Pix p.82**

対になった画像の組から、変換の仕方を学習するGANです。線画から写真、白黒から着色、とい…

**PoC p.93**

概念実証と訳されます。本格的に作り始める前に、そもそも実現できるのかを小さく試して確かめ…

**PoC p.93**

概念実証のことです。契約を段階に分ける進め方では、最初の段階として置かれます。ここで実現…

**Python p.93**

機械学習で最もよく使われているプログラミング言語です。文法が読みやすく、数値計算や機械学…

**Question-Answering p.15**

質問応答と訳されます。検索エンジンのように関係しそうな文書を並べるのではなく、質問に対す…

**Q学習 p.29**

ある状態である行動を取ったときの価値、つまりQ値を、経験しながら更新していく手法です。実…

**REINFORCE p.29**

方策勾配法の代表的な手法です。実際に行動してみても得られた報酬をもとに、良い結果につながっ…

**RLHF p.79**

人間が示した好みを報酬として学習させる手法です。どちらの出力が良いかを人が選び、その判断…

**RMSprop p.44**

AdaGradの学習率が下がりすぎる弱点を、過去の勾配を指数的に忘れていくことで改善した…

**ROC曲線・AUC p.31**

判定のしきい値を動かしたときに、真陽性率と偽陽性率がどう変わるかを描いた曲線がROC曲線…

**Rainbow p.79**

DQNに対して提案されてきた改良手法を、七つまとめて組み合わせたものです。ダブルDQNや…

**RandAugument p.62**

あらかじめ用意した複数の変換の中から、ランダムに選んで適用する方法です。どの変換をどれだ…

**Random Erasing p.62**

画像の中の、ランダムな位置と大きさの領域を消してしまう方法です。対象の一部が隠れた状況を…

**Random Flip p.62**

画像を左右や上下に反転させて水増しする方法です。ただし常に使えるわけではありません。文字…

**ReLU関数 p.39**

入力が0以下なら0を、正ならその値をそのまま返す活性化関数です。正の側で勾配が常に1のま…

**ResNet p.55**

層をいくつか飛び越えて入力をそのまま足す、スキップ結合を導入したネットワークです。これに…

**ResNet p.55**

層を飛び越えて入力を足すスキップ結合によって、非常に深いネットワークの学習を可能にしたモ…

**Rotate p.62**

画像を回転させて水増しする方法です。傾いて写っていても認識できるようになります。ただしこ…

**SARSA p.30**

実際に次に取った行動を使って価値を更新する手法です。自分が現に従っている方策に沿って更新…

**SENet p.69**

チャンネルごとの重要度を学習し、重み付けする仕組みを加えたモデルです。全体を一度圧縮してか…

**SHAP p.87**

それぞれの特徴が予測にどれだけ寄与したかを、協力ゲーム理論のシャープレイ値を使って公平に…

**SHRDLU p.13**

テリー・ウィノグラードが1970年前後に作った、積み木の世界を扱うシステムです。「赤い積…

**SSD p.70**

領域の候補をあらかじめ作らず、一度の処理で検出まで済ませる物体検出モデルです。解像度の異…

**STRIPS p.13**

目標に至る行動の手順を組み立てる、プランニングの手法です。1971年にリチャード・ファイ…

**SaaS p.112**

ソフトウェアを、購入して手元に置くのではなく、ネットワーク越しに使う形で提供する仕組みで…

**SegNet p.69**

エンコーダで小さくし、デコーダで元の大きさへ戻す構造のセマンティックセグメンテーションの…

**Self-Attention p.58**

同じ系列の中で、それぞれの要素が他のすべての要素との関係を見る仕組みです。離れた語どうし…

**Seq2Seq p.59**

入力の系列を一度まとめるエンコーダと、そこから出力の系列を作るデコーダを組み合わせた枠組…

**Seq2Seq p.59**

入力の系列をエンコーダでまとめ、デコーダで別の系列に変換する仕組みです。機械翻訳や文書要…

**Source-Target Attention p.59**

デコーダ側が、エンコーダ側の出力に注目する形のAttentionです。翻訳でいえば、出力…

**TF-IDF p.73**

語の重要度を測る指標です。その文書の中でよく出てくる語ほど重要とし、同時に、どの文書にも…

**TPU p.38**

Googleがディープラーニング専用に開発した演算装置です。行列演算に用途を絞り込むこと…

**Text-To-Image p.85**

文章の指示から画像を作り出す課題です。DALL-EやStable Diffusionが代…

**Transformer p.59**

回帰結合を一切使わず、Attentionだけで構成したモデルです。2017年の論文「At…

**Triplet Loss p.40**

基準となるデータ、同じ仲間のデータ、違う仲間のデータの三つを一組にして使う誤差関数です。…

**U-Net p.70**

エンコーダとデコーダをU字型に配置し、対応する層どうしを直接つないだモデルです。細かい位…

**UCB方策 p.29**

これまでの平均的な良さに加えて、試した回数の少なさも足し合わせて選ぶ方策です。あまり試し…

**Unified-IO p.85**

画像、文章、領域の指定といった多様な入力と出力を、一つのモデルでまとめて扱えるようにした…

**VGG p.70**

3×3という小さなフィルタだけを積み重ねた、単純で分かりやすい構造のモデルです。小さいフ…

**VQ-VAE・info VAE・ $\beta$ -VAE p.60**

変分オートエンコーダを発展させたモデルです。VQ-VAEは潜在変数を連続ではなく離散の値…

**Vision Transformer p.70**

画像を小さな区画に分割し、それを並んだ要素とみなしてTransformerで処理するモデ…

**Visual Question Answering p.85**

画像を見せたうえで、その画像についての質問に答える課題です。写真に何人写っているか、とい…

**WaveNet p.76**

DeepMindが開発した音声合成のモデルです。音の波形そのものを、一点ずつ順に直接作り…

**Web API p.93**

作った機能を、ネットワーク越しに呼び出せる形で提供する仕組みです。学習したモデルをこの形…

**Wide ResNet p.70**

ResNetを深くする代わりに、各層のチャンネル数、つまり幅を広げたモデルです。深さだけが…

**YOLO p.70**

画像を一度見るだけで物体の位置と種類を同時に求める検出モデルです。名前はYou Only…

**Zero-shot p.85**

例を一つも与えずに、指示だけで新しい課題をこなすことです。事前学習で身につけた広い知識が…

**fastText p.72**

word2vecを発展させた手法です。語をそのまま扱うのではなく、文字のまとまりに分解し…

**is-aの関係・has-aの関係・part-ofの関係 p.15**

意味ネットワークで、概念どうしを結ぶ代表的な関係です。is-aは「～の一種である」という…

**k-means法 p.26**

データをk個のかたまりに分ける、代表的な非階層クラスタリングです。まず重心をk個置き、各…

**k-分割交差検証 p.30**

データをk個のかたまりに分け、そのうち一つを検証用、残りを学習用として評価し、これをk回…

**n-gram p.73**

連続するn個の語、あるいは文字のまとまりを指します。二つならバイグラム、三つならトライグ…

**noising p.62**

データにわざと雑音を加えて水増しする方法です。多少の乱れがあっても正しく判断できるように…

**paraphrasing p.62**

文章を、意味を変えずに別の言い方に書き換えて水増しする方法です。自然言語処理におけるデー…

**sim2real p.80**

シミュレータの中で学習した方策を、現実の機械へ移すことです。現実で試すのは費用も危険も大…

**t-SNE p.26**

高い次元のデータを二次元などに落として、目で見えるようにする次元削減の手法です。もとの空…

**tanh関数 p.39**

マイナス1から1までの値を返す、S字型の活性化関数です。0から1を返すシグモイド関数に比…

**word2vec p.73**

語を、意味の近さが計算できるベクトルに変換する手法です。似た文脈で使われる語は似たベクトル…

 **$\alpha\beta$ 法 p.13**

Mini-Max法の読みを効率よくする方法です。途中まで読んだ時点で、これ以上調べても結…

 **$\epsilon$ -greedy方策 p.29**

ほとんどの場合は今いちばん良いとされる行動を選び、ある確率  $\epsilon$  だけ、あえて別の行動を試す方…

**アジャイル p.94**

短い期間で作っては試すことを繰り返しながら進める開発の進め方です。AIは、やってみないと…

**アノテーション p.95**

データに正解のラベルを付けていく作業です。教師あり学習には欠かせませんが、大量に行うには…

**アルゴリズムバイアス p.119**

AIの判断に、特定の属性の人が不利になるような偏りが生じることです。学習データに含まれる…

**アルファスター(AlphaStar) p.90**

DeepMindが開発した、StarCraft IIというゲームのエージェントです。盤面…

**アルファ碁(AlphaGo) p.21**

DeepMindが開発した囲碁のプログラムです。2016年に世界トップ級の棋士イ・セドル…

**アンサンブル学習 p.23**

複数のモデルを組み合わせ、一つのモデルより良い結果を得ようとする方法です。一つひとつは…

**イテレーション p.45**

重みを一回更新することを指します。ミニバッチ学習では、一つのミニバッチを処理して更新する…

**イライザ(ELIZA) p.20**

1960年代にジョセフ・ワイゼンバウムが作った対話プログラムです。相手の言葉を型に当ては…

**インクルージョン p.123**

多様な人々を排除せず、誰もが使える形にするという考えです。特定の属性の人だけ精度が落ち…

**インスタンスセグメンテーション p.71**

画素ごとに塗り分けたうえで、同じ種類の物体でも一つひとつを別のものとして区別する手法です…

**インスタンス正規化 p.54**

一つのデータの、一つのチャンネルごとに正規化する方法です。画像それぞれが持つ明るさや色合い…

**インタビューシステム p.16**

専門家に質問を重ねることで知識を引き出し、ルールの形で取り込む仕組みです。エキスパートシ…

**ウェブマイニング p.16**

ウェブ上のページの内容や、利用者の閲覧履歴から、役に立つ知識を取り出すことです。データマ…

**ウォーターフォール p.94**

要件定義から設計、実装、試験へと、上流から下流へ順に進める開発の進め方です。仕様が最初か…

**ワード法 p.27**

階層クラスタリングの代表的な方法です。二つのかたまりを一つにまとめたときに、内部のばらつ…

**エコーチェンバー p.122**

同じ意見の人ばかりが集まる場で、似た主張が反響し合い、その考えが正しいと強く思い込んでし…

**エッジAI p.87**

手元の機器の中でAIを動かすことです。撮影した映像などを外へ送らずに済むため、通信が要ら…

**エポック p.45**

学習データ全体を一通り学習し終わることを指します。エポック数を増やすほど学習は進みますが…

**エルマンネットワーク p.56**

隠れ層の出力を、次の時刻の入力として戻す形の回帰結合ネットワークです。出力層の値を戻すジ…

**エージェント p.6**

周囲の状況を認識し、自分で判断して行動するプログラムを指します。環境を知覚し、次の一手を…

**オッカムの剃刀 p.32**

同じことを説明できるのであれば、より単純な方を選ぶべきだ、という考え方です。もとは中世の…

**オフライン強化学習 p.80**

あらかじめ集めておいたデータだけを使い、環境と実際にやりとりせずに学習する方法です。試行…

**オントロジー p.16**

概念と概念どうしの関係を、誰が使っても同じ意味になるようにきちんと定義したものです。もと…

**オンライン学習 p.45**

データの一つずつ使って、その都度重みを更新していく方法です。データが届くたびに学び続けら…

**オープンデータセット p.95**

公開されていて誰でも利用できる、データの集まりです。ImageNetなどが代表例で、研究…

**オープン・イノベーション p.94**

自社の中だけで完結させず、外部の知見や技術を取り入れて新しい価値を生み出す考え方です。A…

**カメラ画像利活用ガイドブック p.118**

カメラで撮影した画像を事業に使う際の、実務上の留意点をまとめた指針です。撮影していること…

**カルバック・ライブラー情報量(KL) p.48**

二つの確率分布がどれくらい違うかを測る量です。値は必ず0以上で、二つが完全に同じときだけ…

**カーネル p.23**

データをより高い次元へ写す働きをする関数です。もとの空間では直線で分けられないデータでも…

**カーネル p.23**

畳み込みで使う小さな行列のことで、フィルタとも呼ばれます。注意したいのは、サポートベクタ…

**カーネルトリック p.23**

高い次元へ実際にデータを写す計算をせずに、写した先での内積だけをカーネル関数で求めてしま…

**キー p.60**

Attentionで、照合される側が持つ情報です。クエリと突き合わせて、どれだけ関係が深…

**クエリ p.60**

Attentionで、探しに行く側が持つ情報です。これを各キーと突き合わせて、どれだけ関…

**クラウド p.94**

計算資源をネットワーク越しに借りて使う仕組みです。高価なGPUを自前で持たなくても、必要…

**クラスタリング p.27**

似たもの同士をグループにまとめることです。正解ラベルを与えずに構造を見つけるので、教師な…

**グリッドサーチ p.46**

ハイパーパラメータの候補を格子状に並べ、その組み合わせをすべて試す方法です。網羅的に探せ…

**グループ正規化 p.53**

チャンネルをいくつかの組に分け、その組ごとに正規化する方法です。バッチ正規化とレイヤー正規…

**グローバルアベレージプーリング(GAP) p.64**

特徴マップ全体の平均を取って、一つの値にまとめる方法です。最後の全結合層の代わりに使うこ…

**ゲート機構 p.57**

情報をどれだけ通し、どれだけ遮るかを、学習によって制御する仕組みです。LSTMやGRUの…

**コサイン類似度 p.99**

二つのベクトルの向きがどれくらい近いかを測る指標です。マイナス1から1までの値を取り、1…

**コンテンツベースフィルタリング p.27**

すすめる対象そのものの特徴が似ているものを推薦する方法です。同じジャンルや同じ作者、とい…

**コーパス p.95**

自然言語処理で使うために集められた、大量の文章のデータです。単に集めただけでなく、品詞や…

**コールドスタート問題 p.27**

新しい利用者や新しい商品は履歴がないため、協調フィルタリングでは推薦ができない、という問…

**サポートベクターマシン(SVM) p.35**

データを分ける境界を、最も近いデータとの余白が最大になるように引く手法です。この考え方を…

**サンプリングバイアス p.119**

データの集め方に偏りがあるために、全体を正しく反映しなくなることです。特定の地域や年代か…

**シグモイド関数 p.40**

0から1までの値を返す、S字型の活性化関数です。出力を確率として扱えるため、二値分類の出…

**シンギュラリティ p.7**

技術的特異点と訳されます。AIが自分より賢いAIを作れるようになり、その連鎖によって、人…

**シンボルグラウンディング問題 p.8**

記号と、それが指す実世界の意味を、どう結びつけるかという問題です。たとえばコンピュータが…

**ジョルダンネットワーク p.57**

出力層の出力を、次の時刻の入力として戻す形の回帰結合ネットワークです。隠れ層の出力を戻す…

**スキップグラム p.74**

word2vecの学習方式の一つです。真ん中の語から、その周りに現れる語を当てさせること…

**スキルの喪失 p.123**

AIに任せきりにすることで、人がもともと持っていた技能を失ってしまうことです。異常が起き…

**ステークホルダーのニーズ p.94**

利害関係者が何を求めているかを把握することです。経営層、現場の担当者、そして利用者では、…

**ストライド p.52**

フィルタを一度にどれだけずらすかという幅です。1なら1画素ずつ、2なら2画素ずつ動かしま…

**スパムフィルタ p.17**

迷惑メールを自動で見分ける仕組みです。人が条件を書くのではなく、過去に届いたメールから見…

**スペクトル包絡 p.77**

周波数ごとの強さを並べたとき、その山の連なりをなめらかにつないだ線のことです。声道の形に…

**セキュリティ・バイ・デザイン p.120**

設計の段階から安全性を組み込むという考え方です。作ってから対策を追加するのでは、守り切れ…

**セマンティックWeb p.16**

ウェブ上の情報に意味を表す情報を付け、コンピュータが内容そのものを扱えるようにしようとい…

**セマンティックセグメンテーション p.71**

画像の画素一つひとつが、どのカテゴリに属するかを判定して塗り分ける手法です。ただし同じ種…

**センシティブ属性 p.119**

人種、性別、信条、病歴など、差別につながりかねない属性のことです。公平性を考えるうえで、…

**ソフトマックス関数 p.40**

複数の出力を、合計がちょうど1になるように変換する関数です。それぞれの値を、そのクラスで…

**ソフトロー p.118**

指針や行動規範のように、法的な強制力を持たない決まりのことです。状況の変化に合わせて素早…

**ダイバーシティ p.124**

多様性のことです。開発に関わる人が偏っていると、気づかれない偏りがそのまま製品に入り込み…

**ダブルDQN p.80**

どの行動を選ぶかと、その行動の価値をいくらと見るかを、別々のネットワークで行う手法です。…

**ダートマス会議 p.8**

1956年に、アメリカのダートマス大学で開かれた研究会です。ジョン・マッカーシーが、この…

**チューリングテスト p.9**

機械が知能を持つといえるかどうかを判定する方法です。アラン・チューリングが1950年に提…

**ディープフェイク p.121**

ディープラーニングを使って作られた、本物と見分けがつかない偽の映像や音声です。有名人に言…

**ディープラーニング p.7**

層を深く重ねたニューラルネットワークを使う、機械学習の一つの方法です。それまでの機械学習…

**デュエリングネットワーク p.80**

状態そのものの良さと、その状態でどの行動が優れているかという二つを、分けて学習する構造で…

**デンドログラム(樹形図) p.28**

階層クラスタリングの結果を木の形で表した図です。どのデータがどの順に併合されたかが縦の高…

**データの偏り p.119**

学習に使うデータが、現実の分布を正しく映していない状態です。社会に元からある偏りが反映さ…

**データの来歴 p.121**

そのデータがどこから来て、どう加工されてきたかという履歴のことです。学習に使ったデータの…

**データサイエンティスト p.95**

データから価値を引き出すことを担う専門職です。統計や数理の力、情報技術の力、そして課題を…

**データマイニング p.16**

大量のデータの中から、それまで気づかれていなかった規則やパターンを掘り出すことです。鉱山…

**データリークage p.95**

本来なら使えないはずの情報が学習データに紛れ込み、見かけ上の精度が実際より高く出てしまう…

**データ利用権 p.112**

誰がどのデータをどのように使えるかという取り決めです。データには所有権という考え方がその…

**データ汚染 p.120**

学習に使うデータに、悪意のあるデータを混ぜ込む攻撃です。特定の入力にだけ誤った反応をする…

**データ窃取 p.120**

学習に使われたデータを、モデルの出力から推測して盗み出す攻撃です。ある人のデータが学習に…

**トイ・プロブレム p.8**

迷路や積み木、パズルのように、規則が単純で、扱う範囲が限られた問題のことです。第一次AI…

**トピックモデル p.28**

文書の背後にある話題を推定する、教師なし学習のモデルです。あらかじめ分類先を決めなくても…

**トレーサビリティ p.125**

どのデータを使い、どのモデルで、いつ、どんな判断をしたのかを、後からたどれるようにしてお…

**ドメインランダムマイゼーション p.81**

シミュレータの見た目や物理の条件をランダムに変えながら学習させる方法です。さまざまな条件…

**ドロップアウト p.42**

学習のたびに、一部のノードをランダムに使わないようにする手法です。毎回少しずつ違う形のネ…

**ネオコグニトロン p.19**

1979年に福島邦彦が提案した、人間の視覚のしくみをまねた多層のモデルです。単純な特徴を…

**ノイジーネットワーク p.81**

ネットワークの重みそのものに雑音を加えることで、行動にばらつきを生ませる手法です。 $\epsilon$ の確…

**ノーフリーランチの定理 p.46**

あらゆる問題に対して万能に優れた手法は存在しない、という定理です。ある問題で優れた手法は…

**ハイパーパラメータ p.46**

学習によって決まるのではなく、人があらかじめ決めておく設定値です。学習率、層の数、ミニバ…

**ハノイの塔 p.14**

3本の棒と大きさの違う円盤を使うパズルです。円盤を1枚ずつ動かし、大きい円盤を小さい円盤…

**ハードロー p.117**

法律や規則のように、国家によって強制力を持って定められた決まりのことです。違反すれば罰則…

**バギング p.24**

学習データから重複を許して選び直したものでモデルを複数作り、多数決や平均を取る方法です。…

**バッチ学習 p.47**

学習データ全体を使って誤差を計算し、一回だけ重みを更新する方法です。更新の向きが安定する…

**バッチ正規化 p.53**

ミニバッチの中で、平均が0、分散が1になるように整える方法です。学習が速く安定し、勾配消…

**バリュー p.60**

Attentionで、実際に取り出される値です。クエリとキーの照合で求めた重みを使って、…

**バンディットアルゴリズム p.30**

複数の選択肢のうちどれを選ぶかを、活用と探索を両立させながら決める枠組みです。当たりの確…

**パディング p.52**

入力の周りを0などで埋めてから畳み込むことです。埋めないと端の画素が中心になる機会が少な…

**パノプティックセグメンテーション p.71**

セマンティックセグメンテーションとインスタンスセグメンテーションを統合した手法です。空や…

**パルス符号変調器(PCM) p.90**

アナログの音をデジタルに変換する、代表的な方式です。標本化、量子化、符号化という三つの段…

**ビッグデータ p.17**

量が多く、種類も多様で、発生する速度も速いデータのことです。インターネットとセンサーの普…

**フィルタ p.53**

畳み込みで画像の上を滑らせる小さな行列で、カーネルと同じものです。どんな特徴に反応するか…

**フィルターバブル p.122**

推薦の仕組みが利用者の好みに合う情報ばかりを見せるため、それ以外の情報に触れられなくなる…

**フェイクニュース p.121**

事実でない情報を、本当のニュースのように見せかけて広めるものです。生成AIによって、大量…

**フェイクニュース p.121**

民主主義との関係では、偽の情報が選挙や世論に影響を与えることが問題になります。生成AIに…

**フォルマント p.77**

スペクトル包絡に現れる、いくつかの山のことで。声道の共鳴によって、特定の周波数が強めら…

**フォルマント周波数 p.77**

フォルマント、つまりスペクトル包絡の山が現れる周波数のことです。低い方から順に、第一フォ…

**フレーム問題 p.10**

ある行動をするときに、関係のあることだけを取り出して考えるのが難しい、という問題です。ジ…

**ブラックボックス p.122**

中で何が起きているのかが分からない状態を指します。ディープラーニングは層が深く、判断の理…

**ブルートフォース p.14**

力任せの探索とも呼ばれ、すべての場合をしらみつぶしに調べる方法です。答えがあれば必ず見つ…

**ブースティング p.25**

前のモデルが間違えたデータを重く見ながら、モデルを順番に作っていく方法です。誤りを一つず…

**ブートストラップサンプリング p.25**

もとのデータから重複を許して、同じ数だけ選び直すことです。選ばれないデータも出るため、モ…

**プライバシー・バイ・デザイン p.118**

あとから対策を足すのではなく、設計の段階からプライバシーへの配慮を組み込むという考え方で…

**プルーニング p.88**

重要でない重みや結合を削り取って、モデルを小さくする手法です。枝を刈り込むことにたとえた…

**ベイズ情報量規準(BIC) p.36**

赤池情報量規準と同じく、当てはまりと複雑さの両方からモデルを選ぶ基準です。データの数が多…

**ベクトル自己回帰モデル(VARモデル) p.25**

複数の時系列が互いに影響し合う形に広げた自己回帰モデルです。金利と株価のように、一方の過…

**ベルヌーイ分布 p.102**

成功か失敗かという、二つの結果しか取らない試行を一回だけ行ったときの分布です。コインを一…

**ホールドアウト検証 p.34**

データを学習用と検証用に一度だけ分けて評価する、最も単純な方法です。手順が軽く済む一方で…

**ポアソン分布 p.102**

一定の時間や範囲の中で、まれにしか起きない事象が何回起きるかを表す分布です。一時間あたり…

**マイシン(MYCIN) p.20**

1970年代にスタンフォード大学で作られたエキスパートシステムです。血液の細菌感染症を診…

**マハラノビス距離 p.102**

データの散らばり方と、変数どうしの相関を考慮して測る距離です。単純な直線距離では近く見え…

**マルコフ決定過程 p.30**

次にどの状態へ移るかが、今の状態と今取る行動だけで決まるとみなす枠組みです。それより前の…

**マルチエージェント強化学習(MARL) p.90**

複数のエージェントが同じ環境の中で、同時に学習する枠組みです。互いの行動が環境を変え…

**マルチタスク学習 p.86**

複数の課題を、一つのモデルで同時に学習させる方法です。課題どうしで共通する知識を分け合…

**マージン最大化 p.25**

分ける境界と、それに最も近いデータとの距離を最大にするという考え方です。この距離をマージ…

**ミニバッチ学習 p.47**

学習データを小さなまとりに分け、そのまとりごとに更新する方法です。バッチ学習の安定さ…

**メル周波数ケプストラム係数(MFCC) p.78**

人の耳の聞こえ方に合わせて、音声の特徴を数値の並びに直したものです。声の高さの影響を抑え…

**メル尺度 p.78**

人が実際に感じる音の高さに合わせて作られた尺度です。人の耳は低い音の違いには敏感で、高い…

**モデル圧縮 p.88**

モデルの大きさや計算量を減らす技術の総称です。蒸留、プルーニング、量子化などが含まれます…

**モデル学習の電力消費 p.123**

大規模なモデルの学習には、膨大な計算と、それに伴う大量の電力が必要になります。一度の学習…

**モデル汚染 p.120**

モデルそのものに手を加えて、意図しない振る舞いを仕込む攻撃です。公開されている学習済みモ…

**モデル窃取 p.120**

公開されているモデルに何度も問い合わせ、その入出力からモデルそのものを再現してしまう攻撃…

**モニタリング p.125**

運用を始めたあとも、AIの振る舞いを継続して監視することです。データの傾向が変われば精度…

**モンテカルロ法 p.14**

乱数を使って何度も試し、その結果から良い手を見積もる方法です。最後まで読み切る代わりに、…

**モーメント p.47**

前回の更新の向きを、慣性として今回の更新に加える工夫です。谷を左右に往復するような振動が…

**ユークリッド距離 p.102**

二点を結ぶ直線の長さで測る、最も基本的な距離です。三平方の定理をそのまま多次元に広げた形…

**ラッソ回帰 p.42**

L1正則化を組み込んだ線形回帰です。効いていない説明変数の係数がちょうど0になりやすいた…

**ランダムサーチ p.47**

ハイパーパラメータの組み合わせをランダムに選んで試す方法です。格子状に全部試すグリッドサ…

**ランダムフォレスト p.26**

決定木をバギングで多数作り、さらに各木が使う特徴もランダムに選ぶ手法です。木どうしが似す…

**リカレントニューラルネットワーク(RNN) p.55**

前の時刻の情報を保持しながら、系列を順に処理していくネットワークです。同じ重みを各時刻で…

**リスクベースアプローチ p.118**

AIの用途が持つ危険の大きさに応じて、規制の強さを変えろという考え方です。EUのAI法が…

**リッジ回帰 p.42**

L2正則化を組み込んだ線形回帰です。係数全体を小さく抑えることで、推定が不安定になるのを…

**ルールベース機械翻訳 p.10**

文法の規則と辞書を人が書き、そのとおりに訳す方式です。統計的機械翻訳より前の主流でした。…

**レイヤー正規化 p.53**

一つのデータの中で、すべてのチャンネルをまとめて正規化する方法です。バッチの大きさに全く影…

**レコメンデーションエンジン p.17**

利用者の行動や履歴から好みを推定し、商品や動画をすすめる仕組みです。似た人が好んだものを…

**ロジスティック回帰 p.26**

名前に回帰と付きますが、分類に使う手法です。線形回帰の出力をシグモイド関数に通して0から…

**ローブナーコンテスト p.10**

チューリングテストにもとづいて、最も人間らしい対話ができるプログラムを選ぶコンテストです…

**ワトソン p.17**

IBMが開発した質問応答システムです。2011年にアメリカのクイズ番組ジェパディ！に出場…

**ワンホットベクトル p.75**

語彙の数だけ次元を用意し、その語に当たる場所だけを1、残りをすべて0とする表し方です。作…

**一般物体認識 p.70**

あらかじめ限定された対象だけでなく、日常にある一般的な物体を広く認識することです。特定の…

**不変性の獲得 p.54**

対象の位置が少しずれても、出力が変わらない性質を身につけることです。プーリング層が担う役…

**中国語の部屋 p.9**

ジョン・サールが示した思考実験です。部屋の中の人、マニュアルに従って中国語の文字を並べ…

**中央値 p.100**

データを小さい順に並べたとき、ちょうど真ん中に来る値です。極端に大きな値や小さな値があつ…

**主成分分析(PCA) p.35**

データの散らばりが最も大きい方向を新しい軸として取り直し、次元を減らす代表的な手法です。…

**事前学習 p.60**

層ごとにオートエンコーダとして学習させ、その重みを本番の学習の初期値として使う手法です。…

**事前学習 p.60**

目的の課題に取りかかる前に、大量のデータで先に学習しておくことです。言葉や画像の一般的な…

**事前学習済みモデル p.83**

すでに大量のデータで学習を終えたモデルのことです。これを出発点にして、自分の目的に合わせ…

**二乗平均平方根誤差(RMSE) p.36**

平均二乗誤差の平方根を取った値です。二乗によって変わってしまった単位が、もとのデータと同…

**二重降下現象 p.46**

モデルを大きくしていくと、性能がいったん悪くなった後、さらに大きくすると再び良くなるとい…

**二項分布 p.100**

成功する確率が一定の試行をn回繰り返したときに、成功した回数が従う分布です。コインをn回…

**交差エントロピー p.41**

分類で使われる代表的な誤差関数です。正解のクラスに対してモデルが与えた確率が低いほど、大…

**交差検証 p.32**

データを分けて学習と検証を繰り返し、性能を偏りなく見積もる方法の総称です。一度だけ分ける…

**人工知能 p.6**

人間の知的な働きをコンピュータで実現しようとする技術と、その研究分野のことです。注意した…

**人間の神経回路 p.19**

ニューラルネットワークが手本にした、脳の神経細胞のつながりのことです。神経細胞は、他の細…

**人間の自律性 p.124**

人が自分で考え、自分で決める力のことです。AIによる推薦や助言が強くなりすぎると、知らな…

**人間の関与 p.125**

AIの判断に、人が関わる仕組みを残すことです。人が最終判断を行う形、人が確認して覆せる形…

**他企業や他業種との連携 p.95**

自社だけではデータや知見が足りない場合に、他の企業や別の業種と組むことです。特にデータは…

**代理変数 p.119**

センシティブ属性そのものではないのに、結果としてそれを言い当ててしまう変数のことです。郵…

**仮名加工情報 p.106**

他の情報と照らし合わせない限り、特定の個人が分からないように加工した情報です。組織の内部…

**位置エンコーディング p.59**

語の並び順を表す情報を、入力に足し込む仕組みです。Transformerは系列を並列に処…

**価値原則 p.117**

AIを扱ううえで基礎に置くべき、共通の価値のことです。人間の尊厳、多様性と包摂、持続可能…

**保守契約 p.112**

納品したあとの運用や保守について定める契約です。AIは時間とともに周囲の状況が変わり精度…

**保有個人データ p.107**

事業者が、開示、訂正、利用停止などを行う権限を持っている個人データです。本人から開示など…

**信用割当問題 p.43**

最後に出た結果の良し悪しに対して、どの層のどの重みがどれだけ貢献したのかを割り振る、とい…

**個人データ p.106**

個人情報データベースなどを構成している個人情報のことです。検索できるように体系的にまとめ…

**個人情報 p.106**

生存する個人に関する情報で、特定の個人を識別できるものです。他の情報と容易に照らし合わせ…

**個人識別符号 p.106**

それ単体で特定の個人を識別できる符号のことです。指紋や顔の特徴を変換したデータ、旅券番号…

**倫理アセスメント p.124**

AIを作る前や導入する前に、倫理的な問題が起きないかをあらかじめ評価することです。誰が不…

**偏相関係数 p.101**

第三の変数の影響を取り除いたうえで求めた、二つの変数の相関です。共通の原因によって生じる…

**偽陽性・偽陰性 p.32**

偽陽性は、陽性と判定したのに実際は陰性だった場合です。偽陰性は、陰性と判定したのに実際は…

**公平性の定義 p.119**

何をもって公平とするかは、一つに定まらないという問題です。属性ごとに合格率をそろえるのか…

**公正競争阻害性 p.111**

公正な競争を妨げるおそれがあることです。不公正な取引方法に当たるかどうかを判断する際の基…

**共分散 p.99**

二つの変数が、どれくらい一緒に動くかを表す量です。一方が大きいときにもう一方も大きければ…

**再現性 p.125**

同じ条件で行えば同じ結果が得られることです。学習には乱数が使われるため、条件を記録してお…

**分散 p.101**

それぞれの値が平均からどれだけ離れているかを二乗し、その平均を取ったものです。データの散…

**分散表現 p.74**

語の意味を、多数の次元に分けて表す方法です。一つの次元が一つの意味を担うのではなく…

**分類問題 p.25**

データがどのグループに属するかを当てる問題です。迷惑メールかどうか、どの数字が書かれてい…

**利用目的 p.107**

個人情報何のために使うのかという目的です。取得の際にできる限り特定し、本人に通知するか…

**利用規約 p.108**

サービスを使う際の条件を定めたものです。著作権との関係では、そのサイトの画像や文章を学習…

**利用規約 p.108**

サービスを使う際の条件を定めたものです。AIサービスでは、利用者が入力したデータを学習に…

**割引率 p.30**

将来もらえる報酬を、どれくらい低く見積もるかを決める係数です。0に近いほど目先の報酬を重…

**創作性 p.108**

著作物として保護されるための要件で、作者の個性が何らかの形で表れていることを指します。芸…

**労働力不足 p.123**

人口の減少によって働き手が足りなくなる問題です。AIによる自動化は、これを補う手段として…

**勾配ブースティング p.23**

前のモデルが出した誤差を、次のモデルが埋めるように順番に学習させる、ブースティングの一種…

**勾配消失問題 p.40**

層を深く重ねると、逆向きに伝えていく勾配がだんだん小さくなり、入力に近い層がほとんど学習…

**勾配消失問題 p.40**

誤差逆伝播法の側から見ると、連鎖律にしたがって各段の微分を掛け合わせていくために起こる問…

**勾配消失問題 p.40**

回帰結合層では、時間をさかのぼって誤差を伝えるため、系列が長いほど掛け算の回数が増えます…

**勾配爆発問題 p.43**

勾配消失とは逆に、1より大きい値が何度も掛かって、勾配が発散してしまう問題です。同じ重み…

**勾配爆発問題 p.43**

逆に1より大きい値が重なって、勾配が発散してしまう問題です。同じ重みを何度も掛ける回帰結…

**勾配降下法 p.46**

誤差が小さくなる方向、つまり勾配の逆向きへ、少しずつパラメータを動かしていく方法です。ニ…

**匿名加工情報 p.106**

特定の個人を識別できず、しかも元に戻すこともできないように加工した情報です。この状態にす…

**半教師あり学習 p.84**

少量の正解付きデータと、大量の正解なしデータを組み合わせて学習する方法です。正解を付ける…

**協調フィルタリング p.27**

自分と似た好みを持つ他の利用者の行動をもとに、商品や動画をすすめる方法です。対象そのもの…

**単回帰分析 p.24**

説明変数が一つだけの回帰分析です。たとえば気温だけからアイスの売上を予測する場合が当たり…

**単純パーセプトロン p.39**

1958年にフランク・ローゼンブラットが考案した、最も基本的なモデルです。入力に重みを掛…

**単語埋め込み p.74**

語を、低い次元の実数のベクトルに置き換えて表すことです。ワンホットベクトルのように次元が…

**双方向RNN (Bidirectional RNN) p.65**

系列を前から処理するものと、後ろから処理するものを組み合わせたネットワークです。ある時点…

**営業秘密 p.110**

不正競争防止法で保護される情報です。秘密として管理されていること、事業に有用であること、…

**回帰問題 p.23**

連続する数値を予測する問題です。売上や気温、価格などが対象になります。どのグループに属す…

**基盤モデル p.85**

大量で多様なデータを使って学習し、さまざまな用途に適応させられる大規模なモデルです。一つ…

**報酬成形 p.81**

学習が進みやすいように、途中の段階でも報酬を与えるよう人が設計することです。最終的な成功…

**変分オートエンコーダ(VAE) p.61**

潜在変数を一つの値ではなく、確率分布として学習するオートエンコーダです。分布から値を取り…

**外れ値 p.100**

他のデータから大きく離れた値のことです。測定の誤りである場合もあれば、本当に珍しい現象を…

**多クラス分類 p.24**

三つ以上のグループのどれに当たるかを当てる分類です。手書き数字を0から9のどれかに分ける…

**多層パーセプトロン p.39**

単純パーセプトロンに隠れ層を加えたものです。層を重ねることで直線では分けられない問題、た…

**多次元尺度構成法(MDS) p.35**

データどうしの距離の関係をできるだけ保ったまま、低い次元に配置し直す手法です。もとの特徴…

**大域最適解 p.46**

全体を通して本当に最小となる点です。学習が目指す理想の到達点ですが、複雑なモデルでは、そ…

**大規模言語モデル(LLM) p.75**

膨大な量のテキストで学習した、非常に大きな言語モデルです。パラメータの数とデータ量を増や…

**委託 p.107**

利用目的の達成に必要な範囲で、個人データの取り扱いを外部に任せることです。第三者提供には…

**姿勢推定 p.71**

画像や動画から、人の関節の位置を推定する技術です。肩、肘、手首といった点を求め、それらの…

**学習率 p.45**

一回の更新でどれだけ大きくパラメータを動かすかを決める値です。大きすぎると最小値を通り越…

**宝くじ仮説 p.87**

大きなネットワークの中には、それだけを取り出して学習しても同等の性能を出せる小さな部分ネ…

**対立仮説 p.100**

帰無仮説が否定されたときに、代わりに採用される仮説です。多くは差がある、効果がある、と置…

**局所最適解 p.46**

その周りと比べれば最も低いものの、全体で見れば最小ではない点です。勾配降下法はここに落ち…

**帰無仮説 p.98**

統計的検定において、否定したい方の仮説です。多くは差がない、効果がない、と置きます。この…

**幅優先探索 p.14**

探索木を、根に近い階層から順に、横方向へすべて調べていく方法です。最も浅い所にある答え、…

**平均 p.101**

すべての値を合計して、個数で割ったものです。最もよく使われる代表値ですが、極端に大きな値…

**平均二乗誤差(MSE) p.36**

予測と実際の差を二乗して平均した値で、回帰の評価に使われます。二乗するため、大きく外した…

**平均二乗誤差関数 p.41**

回帰で使われる代表的な誤差関数です。予測と正解の差を二乗して平均します。差を二乗するため…

**平均値プーリング p.55**

決められた範囲の値の平均を取る方法です。最大値プーリングに比べて、範囲全体の情報がなだら…

**平均絶対値誤差(MAE) p.36**

予測と実際の差の絶対値を平均した値です。二乗しないため、大きく外れた一部のデータに引きず…

**度数分布 p.100**

データを値の範囲ごとに区切り、それぞれに何個入るかを数えて並べたものです。表にしたものが…

**強いAIと弱いAI p.10**

ジョン・サールによる区別です。強いAIは、人間と同じように心や理解そのものを持つAIを指…

**形態素解析 p.74**

文を、意味を持つ最小の単位である形態素に分け、それぞれの品詞を判定する処理です。英語と違…

**情報検索 p.74**

大量の文書の中から、求める内容に関係するものを探し出す課題です。検索エンジンが代表例です…

**意味ネットワーク p.15**

知識を書き表す方法の一つです。概念を点、概念どうしの関係を線で表し、網の形にして整理しま…

**感情分析 p.73**

音声から、話し手の感情を読み取る課題です。声の高さ、大きさ、話す速さといった特徴を手が…

**感情分析 p.73**

音声から、話し手の感情を読み取る課題です。声の高さ、大きさ、話す速さといった特徴を手が…

**探索木 p.13**

取りうる場合を木の形に並べ、枝をたどって答えを探す考え方です。根が最初の状態、枝分かれが…

**教師強制 p.57**

学習のときに、前の時刻に自分が出した予測ではなく、正解の値を次の入力として与える方法です…

**敵対的生成ネットワーク(GAN) p.82**

本物らしいデータを作る生成器と、本物か偽物かを見分ける識別器を競い合わせて学習させる仕組…

**文書要約 p.75**

長い文章を短くまとめる課題です。元の文をそのまま抜き出す抽出型と、新しく文を作り直す生成…

**文章生成 p.83**

文章を人工的に作り出す技術です。大規模言語モデルによって、自然で長い文章が作れるようにな…

**新規性 p.109**

まだ世の中に知られていないことという、特許を受けるための要件です。出願の前に論文で発表し…

**方策勾配法 p.30**

価値を見積もるのではなく、方策そのものを勾配にしたがって直接改善していく方法です。行動の…

**早期終了 p.46**

検証用データに対する誤差が下がり止まって上がり始めたところで、学習を打ち切ることです。学…

**時系列データ p.57**

時間の順に並んだデータのことです。株価、気温、音声、文章などが当たります。順番そのものに…

**最大値プーリング p.54**

決められた範囲の中で、最も大きい値だけを取り出す方法です。最も強く反応した場所の情報だけ…

**最小二乗法 p.99**

予測と実際の差を二乗して足し上げたものが、最も小さくなるようにパラメータを決める方法です…

**最尤法 p.99**

実際に観測されたデータが、最も起こりやすくなるようにパラメータを決める方法です。もっとも…

**最頻値 p.99**

データの中で、最も多く現れる値のことです。平均、中央値と並ぶ代表値の一つです。数値でない…

**期待値 p.98**

確率変数が平均的にどれくらいの値になるかを表したものです。取りうる値と、その確率を掛け合…

**条件付き確率 p.99**

ある事象が起きたという条件のもとで、別の事象が起きる確率です。何も分かっていないときの確…

**東ロボくん p.16**

国立情報学研究所が2011年から進めた、東京大学の入試合格を目指すプロジェクトです。多く…

**構文解析 p.74**

文の中で、どの語がどの語に係っているかという関係を明らかにする処理です。形態素解析で切り…

**標準偏差 p.101**

分散の平方根を取ったものです。分散は二乗されているため単位が元と変わってしまいますが、平…

**機械学習 p.7**

人がルールを書き下す代わりに、大量のデータからパターンを学び取らせる方法です。たとえば迷…

**機械翻訳 p.73**

ある言語の文を別の言語へ自動で翻訳する課題です。人が規則を書くルールベース、対訳データか…

**次元の呪い p.17**

扱う特徴の数、つまり次元が増えるほど、学習に必要なデータが爆発的に増えてしまうことです。…

**次元削減 p.27**

扱う特徴の数を減らして、データを小さく扱いやすくすることです。特徴が増えるほど学習が難し…

**次元削減 p.27**

オートエンコーダは、入力をいったん小さな中間層に押し込めてから元に戻すよう学習します。そ…

**正則化 p.42**

モデルが複雑になりすぎるのを防ぐために、パラメータの大きさなどに罰を加える工夫です。学習…

**正規分布 p.99**

平均を中心として左右対称の、釣鐘のような形をした分布です。ガウス分布とも呼ばれます。身長…

**正解率・適合率・再現率・F値 p.33**

正解率は全体のうち正しく判定できた割合です。適合率は陽性と判定したうち本当に陽性だった割…

**死者への敬意 p.124**

亡くなった方の声や姿を、生成AIで再現することをめぐる問題です。遺族の慰めになるという見…

**残差強化学習 p.80**

すでにある制御の仕組みを土台として残し、それとの差分だけを強化学習で学ぶ方法です。ゼロか…

**気候変動 p.123**

AIとの関係では、二つの側面があります。一つは、大規模なモデルの学習が大量の電力を使い、…

**汎化性能 p.33**

学習に使っていない、初めて見るデータに対してどれだけうまく働くかという性能です。機械学習…

**決定木 p.23**

条件による枝分かれを繰り返して予測する、木の形をしたモデルです。「年齢が30以上か」とい…

**深さ優先探索 p.14**

探索木を、行けるところまで一気に下り、行き止まりに当たったら一つ戻って別の枝へ進む方法で…

**混同行列 p.33**

予測と実際の組み合わせを、真陽性・偽陽性・偽陰性・真陰性の四つに整理した表です。正解率、…

**準委任契約 p.111**

一定の業務を行うことを約束する契約で、完成させる義務までは負いません。AI開発は結果が読…

**潜在的ディリクレ配分法(LDA) p.35**

一つの文書が複数の話題の混ざり合いでできている、と考えるトピックモデルの代表的な手法です…

**物体検出 p.71**

画像の中で物体がどこにあるかを四角い枠で示し、あわせてその種類も判定する技術です。位置と…

**物体識別 p.71**

画像に何が写っているかという、種類だけを判定する技術です。どこにあるかという位置は求めま…

**特徴マップ p.52**

畳み込み操作の出力にあたる、二次元の並びです。フィルタが反応した場所ほど大きな値になるた…

**特異値分解(SVD) p.35**

一つの行列を三つの行列の積に分解する方法です。分解したうち重要な成分だけを残せば、もとの…

**特許権 p.110**

発明を独占的に実施できる権利です。著作権と違って、出願して審査を通らなければ発生しません…

**状態価値関数 p.29**

ある状態にいること自体の良さ、つまりその状態から先で得られると見込まれる報酬を表す関数で…

**状態表現学習 p.80**

カメラの映像のような生の観測から、意思決定に役立つ形の表現を学び取ることです。そのまま…

**生成AI p.19**

文章や画像、音声などを新しく作り出すAIの総称です。大量のデータで学んだモデルが、学んだ…

**産学連携 p.94**

企業と、大学や研究機関が協力して研究開発を進めることです。最先端の知見を持つ研究側と、実…

**画像生成 p.82**

学習していない新しい画像を作り出す技術です。GANや変分オートエンコーダ、そして拡散モデ…

**畳み込みニューラルネットワーク(CNN) p.64**

畳み込み層とプーリング層を重ねて作るネットワークです。近くの画素どうしの関係を捉え、位置…

**畳み込み操作 p.52**

フィルタを画像の上で滑らせながら、重なった位置どうしを掛けて足し合わせる操作です。同じフ…

**疑似相関 p.98**

直接の因果関係はないのに、共通の原因があるために関係があるように見えてしまうことです。ア…

**発明 p.109**

自然法則を利用した技術的な思想の創作のうち、高度なものと定義されています。自然法則そのも…

**発明者 p.110**

実際に発明を行った自然人のことです。日本を含む多くの国では、AIは発明者になれないという…

**相互情報量 p.100**

二つの変数が、どれだけ情報を共有しているかを表す量です。片方を知ること、もう片方の不確…

**相関係数 p.99**

二つの変数の間にある直線的な関係の強さを、マイナス1から1までの値で表したものです。共分…

**真陽性・真陰性 p.33**

真陽性は、陽性と判定して実際も陽性だった場合です。真陰性は、陰性と判定して実際も陰性だっ…

**知的財産権 p.109**

知的な創作活動の成果や、営業上の目印に与えられる権利の総称です。特許権、著作権、商標権、…

**知識獲得のボトルネック p.8**

専門家の知識をコンピュータに書き移す作業に、非常に大きな手間がかかることを指します。第二…

**破滅的忘却 p.84**

新しい課題を学習させると、それ以前に学んだ内容を大きく失ってしまう現象です。人間のように…

**確率分布 p.98**

確率変数が、どの値をどれくらいの確率で取るかを表したものです。正規分布や二項分布といった…

**確率変数 p.98**

取る値が確率的に決まる変数のことです。さいころの目や、明日の気温などが当たります。値が一…

**確率密度 p.98**

連続した値を取る確率変数について、ある値の周りにどれだけ確率が詰まっているかを表したも…

**確率的勾配降下法(SGD) p.49**

学習データ全体ではなく、ランダムに選んだ一つ、あるいは少数のデータだけを使って更新する方…

**移動平均 p.98**

一定の区間をずらしながら平均を取っていく方法です。細かい上下の変動がならされ、全体の傾向…

**積層オートエンコーダ p.60**

オートエンコーダを何段も積み重ねたものです。一段ずつ順に事前学習してから、全体をまとめて…

**競争制限 p.111**

市場における競争を妨げることです。AIとの関係では、複数の事業者が同じ価格アルゴリズムを…

**第三者提供 p.106**

個人データを、本人以外の第三者へ渡すことです。原則として、あらかじめ本人の同意を得る必要…

**精度保証 p.112**

一定の精度が出ることを約束することです。AIは学習に使うデータや、実際に入ってくるデータ…

**精度保証 p.112**

サービスを提供する側が、一定の精度を約束することです。AIの出力は入力によって変わるため…

**統計的機械翻訳 p.10**

大量の対訳データから、どの訳し方がどれくらい確からしいかを統計的に学んで翻訳する方式です…

**統計的機械翻訳 p.10**

大量の対訳データから、訳し方の確からしさを統計的に学んで翻訳する方式です。人が規則を書く…

**統計的自然言語処理 p.18**

言葉を統計的に扱い、どの解釈がどれくらい確からしいかで処理する方法です。人が文法の規則を…

**線形回帰 p.24**

データの関係を直線、変数が増えれば平面や超平面で表す、最も基本的な回帰の手法です。予測値…

**線形関数 p.51**

入力に重みを掛けて足し合わせるだけの変換です。全結合層が行っているのはこの計算です。ただ…

**職務発明 p.110**

従業員が職務として行った発明のことです。企業があらかじめ定めておけば権利を企業のものに…

**自己回帰モデル(ARモデル) p.24**

過去の自分自身の値から次の値を予測する、時系列データのモデルです。今日の気温を、昨日や一…

**自己教師あり学習 p.83**

データそのものから正解を作り出して学習する方法です。文章の一部を隠して当てさせる、画像の…

**著作権 p.109**

著作物を独占的に利用できる権利です。著作権法30条の4により、情報解析を目的とする利用は…

**著作権侵害 p.108**

権利者の許しを得ずに著作物を利用することです。判断は、元の著作物に依拠したかという点と、…

**著作物 p.108**

思想または感情を創作的に表現したもので、文芸、学術、美術、音楽の範囲に属するものと定義さ…

**蒸留 p.87**

大きなモデルの出力を教師として使い、小さなモデルに同じ振る舞いを学ばせる手法です。正解ラ…

**行動価値関数 p.29**

ある状態で、ある行動を取ったときに、その後で得られる報酬の見込みを表す関数です。Q値とも…

**要配慮個人情報 p.107**

人種、信条、社会的身分、病歴、犯罪の経歴など、取り扱いに特に配慮を要する情報です。不当な…

**話者識別 p.78**

話しているのが誰かを、声から判定する技術です。何を話したかを問う音声認識とは目的が異なり…

**説明可能AI (XAI) p.87**

なぜその判断に至ったのかを、人が理解できる形で示せるようにする研究分野です。医療や与信の…

**説明可能性 p.121**

AIがなぜその判断をしたのかを、人が理解できる形で示せることです。医療や与信のように、判…

**請負契約 p.111**

仕事を完成させることを約束する契約です。完成しなければ報酬を受け取れません。AIはデータ…

**質問応答 p.74**

自然な文で書かれた質問に対して、答えを返す課題です。関係する文書を並べる情報検索とは違い…

**赤池情報量規準(AIC) p.36**

データへの当てはまりの良さと、使ったパラメータの多さの両方を見て、モデルを選ぶための基準…

**身体性 p.8**

知能は、身体を通して世界と関わることで初めて成り立つ、という考え方です。人間は、触ったり…

**軍事利用 p.124**

AIを兵器や軍事の目的に使うことです。人の判断を介さずに攻撃を決める自律型致死兵器システ…

**連続値制御 p.81**

行動の選択肢が、飛び飛びの選択肢ではなく連続した値になる問題です。ロボットの関節をどの角…

**連鎖律 p.43**

合成関数の微分を、各段階の微分の掛け算として求める規則です。ニューラルネットワークは関数…

**進歩性 p.109**

その分野の通常の技術者が、既に知られている技術から容易に思いつくものではないという要件で…

**過学習 p.32**

学習に使ったデータに合わせすぎてしまい、初めて見るデータに対する精度が落ちてしまうことで…

**重み p.51**

入力にかかる係数のことで、学習によって調整される値そのものです。重みが大きいほど、その入…

**重回帰分析 p.24**

説明変数が複数ある回帰分析です。気温、曜日、天気などを同時に使って売上を予測します。変数…

**量子化 p.88**

重みや計算を、より少ないビット数で表すようにする手法です。たとえば32ビットの小数を8ビ…

**限定提供データ p.110**

特定の相手に限って提供される、電磁的に管理された商業的なデータです。営業秘密の三要件を満…

**隠れマルコフモデル p.76**

直接は観測できない状態の移り変わりを、確率で表すモデルです。音声のように、時間とともに変…

**隠れ層・入力層・出力層 p.38**

ニューラルネットワークを構成する三種類の層です。入力層がデータを受け取り、出力層が結果を…

**鞍点 p.44**

ある方向から見ると谷底なのに、別の方向から見ると山の頂上になっている点です。馬の鞍の形に…

**音声合成 p.76**

文字で書かれた内容から、人が話しているような音声を作り出す技術です。読み上げソフトやカー…

**音声生成 p.82**

音声を人工的に作り出す技術です。文章を読み上げる音声合成のほか、特定の人の声に似せること…

**音声認識 p.76**

人の話した音声を、文字に変換する技術です。かつては隠れマルコフモデルが主力でしたが、現在…

**音素 p.76**

言語の音を分ける、最小の単位です。日本語であれば、母音と子音の組み合わせがこれに当たりま…

**音韻 p.76**

ある言語の中で、意味の違いを生み分ける音の単位です。実際に発せられた物理的な音そのもので…

**高速フーリエ変換(FFT) p.90**

波形を、どんな周波数の成分がどれだけ含まれているかに分解する計算を、高速に行う方法です。…

AI資格ラボ