

aa-7. さまざまな機械学習 －機械学習の広がりを学ぶ

(人工知能)

金子邦彦



機械学習の枠組みと系列データを扱う手法

学習の枠組みのバリエーション

教師あり学習 /
正解ラベルから学習

教師なし学習 /
ラベルなしで学習

教師なし学習：隠れた構造を発見

ラベルの無いデータからパターン・構造・隠れた関係性を見つける
応用：クラスタリング/異常検知/推薦システム/次元削減・データ圧縮

強化学習：行動の結果（報酬・ペナルティ）を手がかりに、より良い行動方針を学習

強化学習 /
経験の出どころ：
外部の環境



自己学習 /
経験の出どころ：
自分自身との対戦

自己学習は強化学習の考え方を応用した発展形

扱うデータの多様性

系列データ（音声・テキストなど）を扱う手法の発展
扱うデータの発展

RNN



LSTM



Transformer / 言語・音声で主流



対話AI

Transformerを利用。入力文から次に来る語を確率で予測し、それを繰り返して応答を生成

RNN・LSTMは順番に処理、Transformerは全体を一度に扱えるため主流に移行

今までの内容と今回の内容の関連



前回までの内容

教師あり学習／画像分類

ニューラルネットワーク（多数のニューロンを層状につなぎ、入力から出力を計算する仕組み）

ラベル（正解）を与えて学ぶ

方向1：学習の枠組みを広げる

教師なし学習
正解のない
データから
パターンを
見つける

強化学習：
報酬・
ペナルティ
(試行錯誤)
から学ぶ

自己学習：
強化学習の
考え方を
応用する

方向2：扱うデータを広げる

画像分類の
ニューラル
ネットワーク、
ディープラーニ
ング（基盤）

系列データ
(順序付けら
れたデータの
並び) へ適用

7-1. 教師なし学習

教師なし学習



ラベル（正解）の無いデータから、パターン・構造・隠れた関係性を見つける手法



クラスタリングの例

顧客の購買パターン分析

異常検知の例

通常データから外れた値の検出

人間が気づいていないパターンを発見できる場合がある

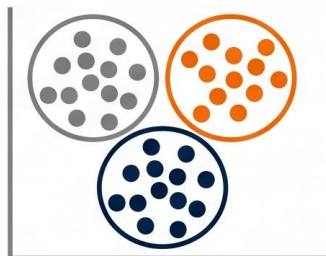
教師なし学習の応用範囲



クラスタリング

似た特徴を持つデータをグループ化する

購買行動に基づく
顧客グループ分け



異常検知

通常と異なるパターンを見つけ出す

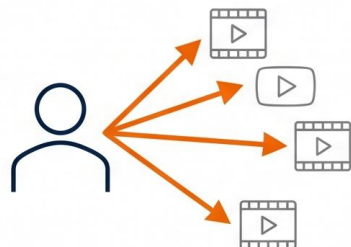
クレジットカードの
不正利用の検出



推薦システム

好みの傾向から関連する項目を提示する

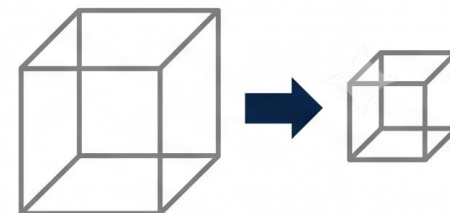
映画配信サービスの
映画推薦



次元削減・データ圧縮

重要な情報を保ちつつ少ない次元で表現する

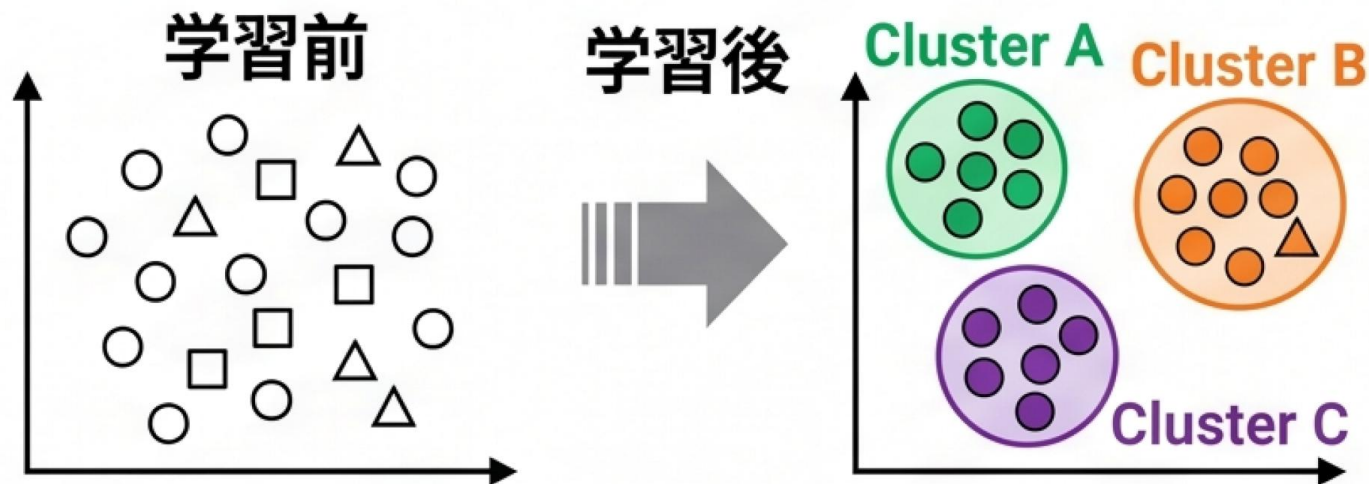
高解像度画像の
圧縮



教師なし学習の応用例① クラスタリング



似ているデータをグループ（クラスター）に分ける

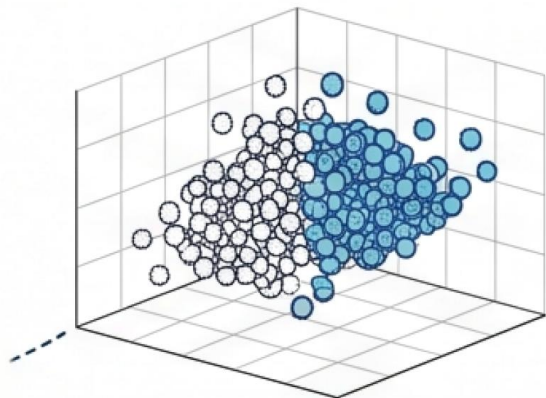


例

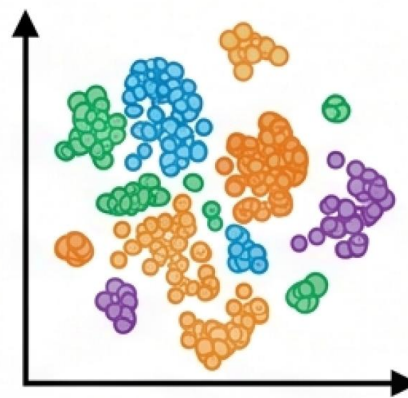
- 顧客セグメンテーション
- 画像分割

例 顧客セグメンテーション・画像分割

多数の特徴量（高次元）を重要な特徴量（低次元）に圧縮する



高次元データ



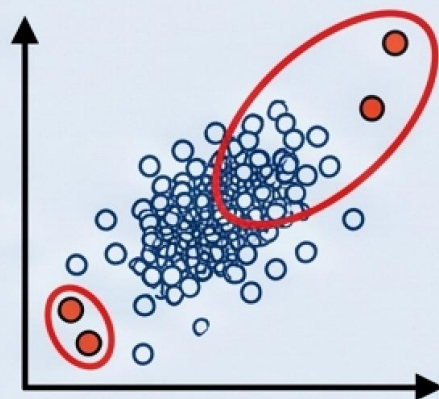
低次元データ

例

- データの可視化
- 特徴量抽出
- ノイズ除去

異常検知

通常のパターンから外れた異常データを特定する



例

- 不正利用検知
- 機器の故障診断

アソシエーションルール学習

データ間の相関関係や規則性を発見する



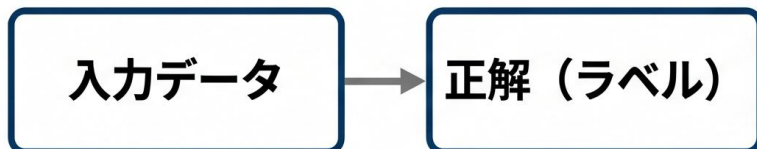
例

- レコメンデーション

教師あり学習と教師なし学習の比較

教師あり学習

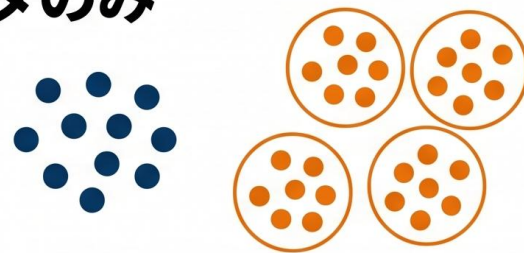
- 使用データ：入力データと正解（ラベル）のペア



- 用途：予測・分類
- 例：迷惑メールの分類
- 特徴：正解が明確である

教師なし学習

- 使用データ：ラベルのないデータのみ



- 用途：パターン発見
- 例：顧客のグループ分け
- 特徴：正解は明確に定まらない

7-2. 学習のバリエーション

機械学習のバリエーション – フィードバックで学ぶ仕組み



教師あり学習

正解データから学ぶ

教師なし学習

正解なしでデータの構造を見つける

フィードバックで学ぶ
報酬やペナルティから学ぶ

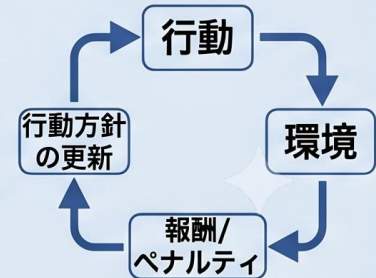
《フィードバックで学ぶ》

自己学習

強化学習の考え方を
応用した発展的な形

強化学習

行動の結果として得られる
報酬・ペナルティを手がかりに、
より良い行動方針を学習する



強化学習のイメージ

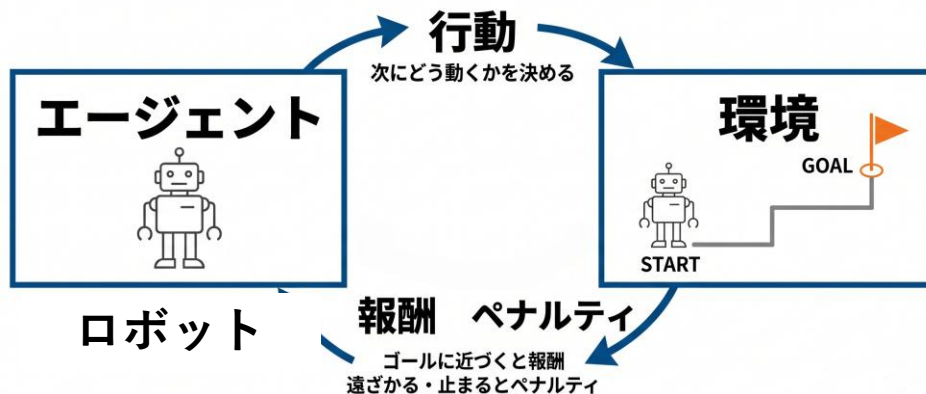
ロボットの学習



ゲームの学習



ゴールを目指すロボット



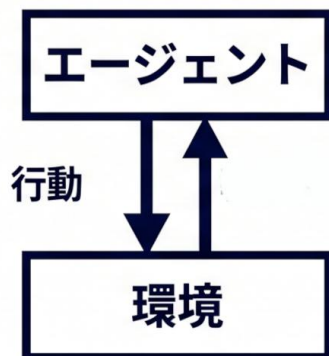
ロボットは行動を繰り返し、
得られた報酬・ペナルティ
を手掛かりに、より良い行
動を学習

定義

フィードバック（報酬やペナルティ）に基づく学習

仕組み

環境との相互作用を通じてデータを収集する



主な特徴

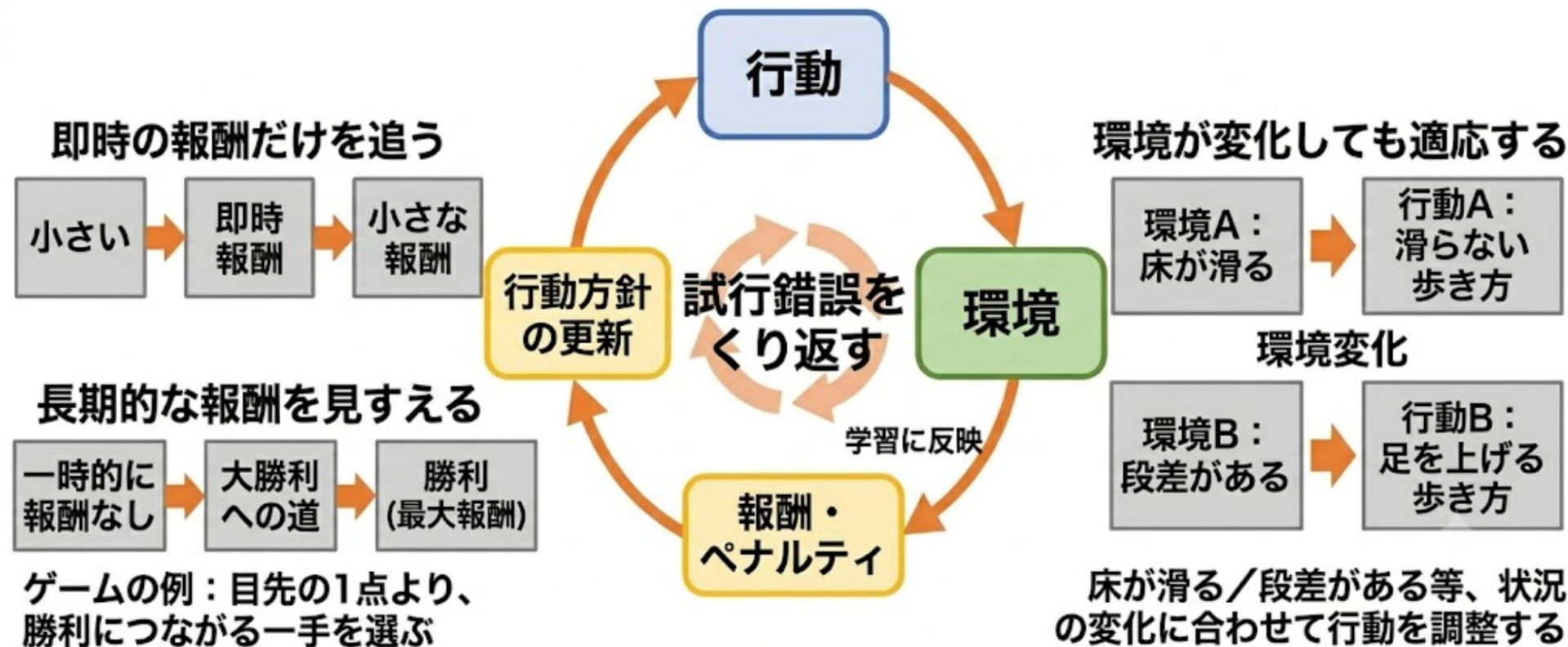
長期的最適化：
即時の報酬だけでなく、長期的なフィードバックをもとに学習する

動的環境対応：
環境の変化に対応する

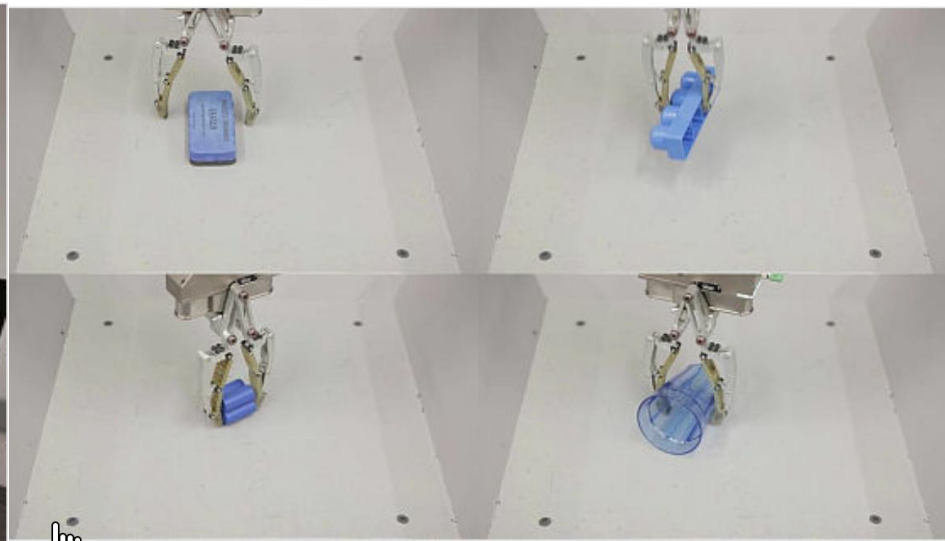
用途

- ゲーム
- ロボット
- 対話AI

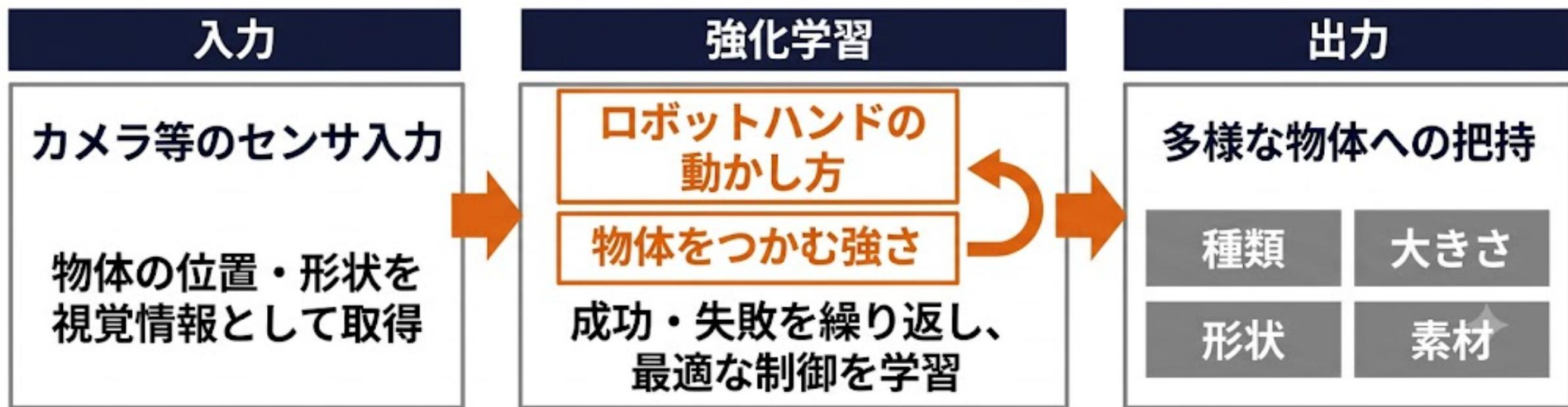
強化学習の仕組み



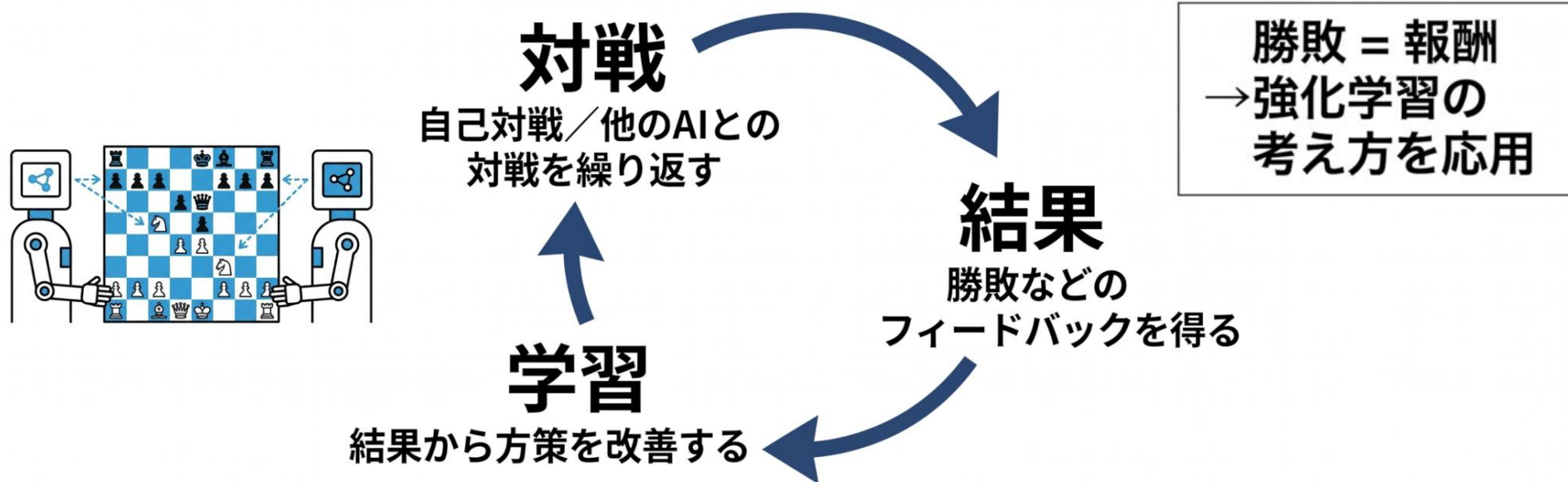
Google のロボットによる強化学習 (2016年) : 多様な物体の把持の学習



<https://www.youtube.com/watch?v=Q9tDHuidzak> より



AIの自己学習



AIが対戦を繰り返し、その結果から学習する手法

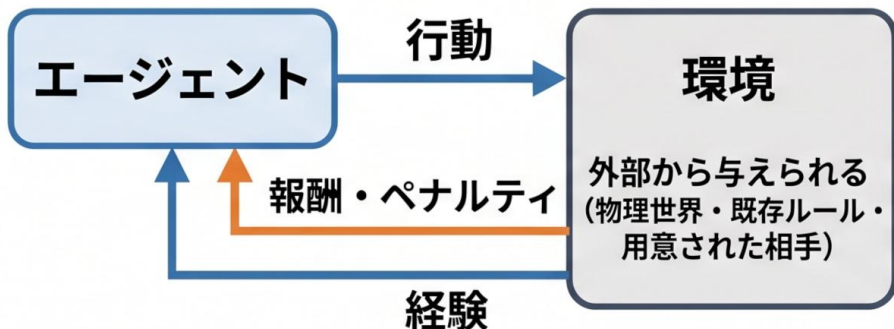
具体例：AlphaZero（2017年）

4時間の学習で、それまで最強だったチェスのプログラムに勝利した

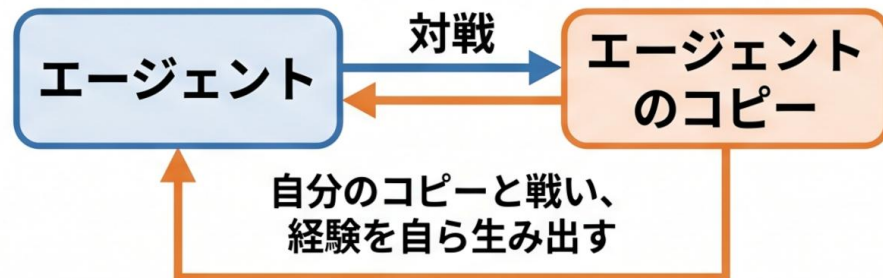
強化学習と自己学習の違い



強化学習



自己学習



自分が強くなると相手も同じだけ強くなる→常に適切な手強さ

項目	強化学習	自己学習
経験の出どころ	外部の環境	自分自身との対戦
相手・課題	外から与えられる	自分で生み出す
相手の強さ	固定されやすい	成長に合わせて強くなる
例	ロボットの把持学習	AlphaZero

自己学習は手本がなくてもゼロから学習を進め、人間を超える水準に到達できる。

自己学習で人間に勝利したAI

OpenAIの人工知能が、数ヶ月の学習でオンラインゲーム
Dota 2 の人間チャンピオンに勝利 (2018年)

ポイント：人間の手本に頼らず、経験から自分で上達する仕組み

① 自己対戦をくり返す

AI同士が対戦し、勝敗から学ぶ。教師となる人間の手本は不要。

② 数ヶ月の高速学習

膨大な対戦を超高速でこなし、人間の何百年分もの経験を短期間で蓄積する。

③ 人間チャンピオンに勝利

Dota 2 のトッププレイヤーを破る (2018年)



従来の学習

人間が用意した
正解データを教わる

自己学習

AIが自ら対戦をくり返し、
自分で強くなる

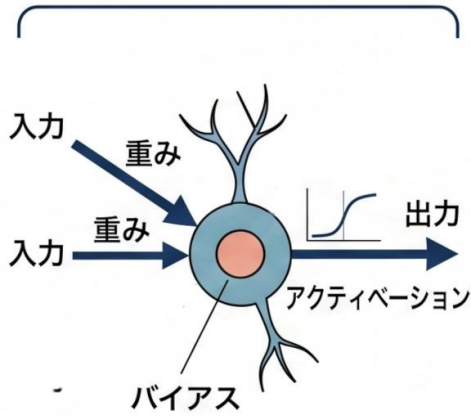


7-3. ディープラーニング の特徴と応用

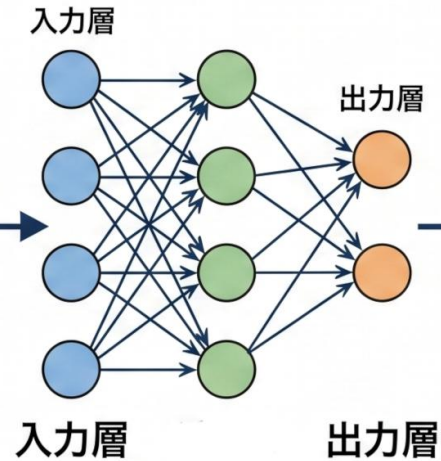
ニューラルネットワークとディープラーニング



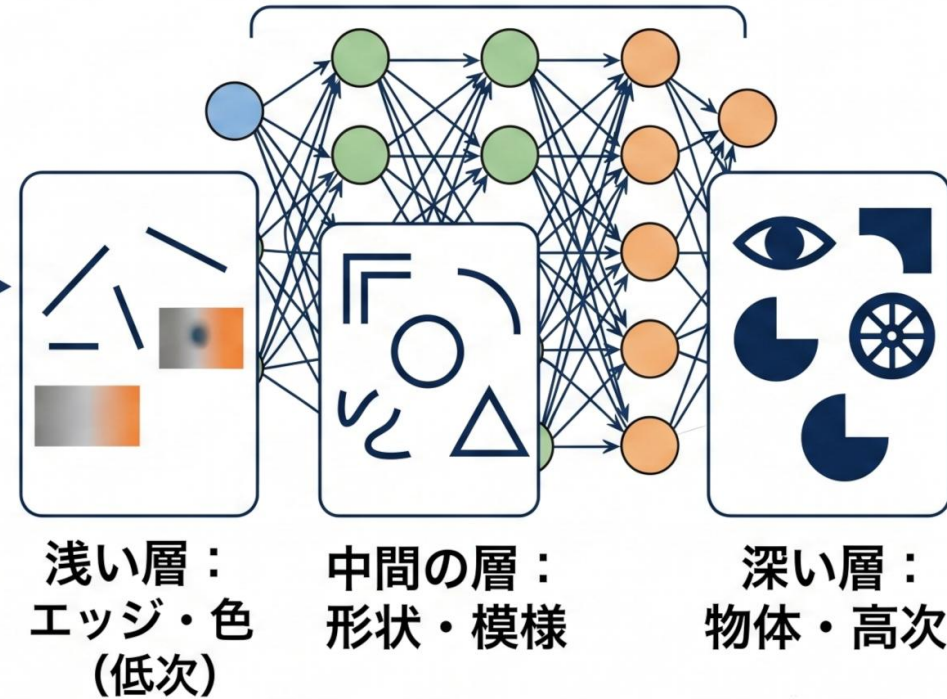
1. ニューロン



2. ニューラルネットワークの層



3. ディープラーニング



ニューラルネットワークは、層が深くなるほど単純なパターンを組み合わせることで複雑なパターンをとらえる



エッジ・色の変化など単純で局所的なパターン（低次の特徴）

組み合わせる



エッジを組み合わせた形状・模様パターン

さらに組み合わせる



形状を組み合わせた物体の部分など複雑なパターン（高次の特徴）

単純・局所的

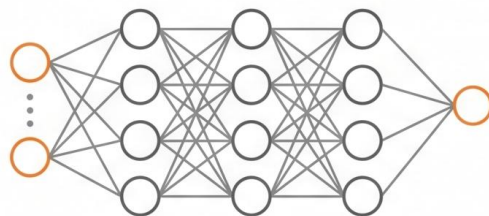
層が深くなる



複雑・全体的

1 「複雑なパターンを抽出する能力」

入力 → 低次の特徴 → 中間の特徴 → 高次の特徴



多層構造により低次から高次へ階層的に特徴を獲得する

2 「多様なデータ形式への対応」

画像
テキスト
音声
動画



ニューラルネットワーク
(同一の枠組み)

画像	→	CNN
系列データ (テキスト・音声)	→	RNN / LSTM
全般 (近年の主流)	→	Transformer

ディープラーニングの応用分野



応用分野

画像認識

物体検出、顔認証、医用画像診断、自動運転の認識系。

自然言語処理

機械翻訳、文章生成、要約、対話システム（大規模言語モデル）。

音声認識

音声入力、文字起こし、音声合成（TTS）。

横断的応用

マルチモーダルモデル（画像＋テキストなど複数データを統合）。

入力：多様なデータ

画像

テキスト

音声

動画

ディープラーニング

複雑なパターンを自動で抽出
パターン抽出とタスク実行を自動化

ディープラーニングは、教師あり学習，教師なし学習，強化学習のすべてで利用されている

学習の枠組み（目的）＝何を手がかりに学ぶか

	教師あり学習 正解ラベルから学ぶ	教師なし学習 ラベルなしデータの構造を学ぶ	強化学習 試行錯誤と報酬から学ぶ
ディープラーニング	●	●	●
従来手法 (線形モデル・決定木など)	●	●	●

実現手段 || どう学習を行うか



系列データ分析

系列データ

順序付けられた、データの並び



- 並びに順序がある / 並べ替えると意味が変わる
- 音声、テキストも系列データ

ニューラルネットワークの再帰（リカレント）の仕組み

フィードフォワード型



ある層の出力を、次の層が受け取る。
情報は一方向にのみ流れる。

RNN型（リカレント）

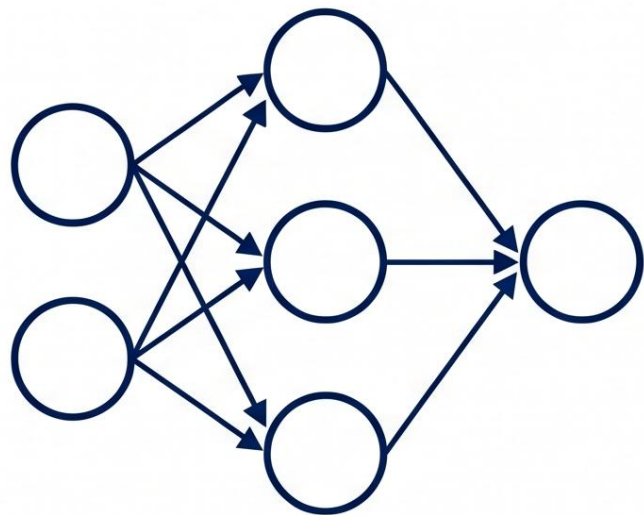


再帰により、前回の実行結果の一部が
次の実行へ引き継がれる。

RNNは並び順を持つ系列データ(テキスト・音声・時系列)に適する

ニューラルネットワークの再帰（リカレント）の仕組み

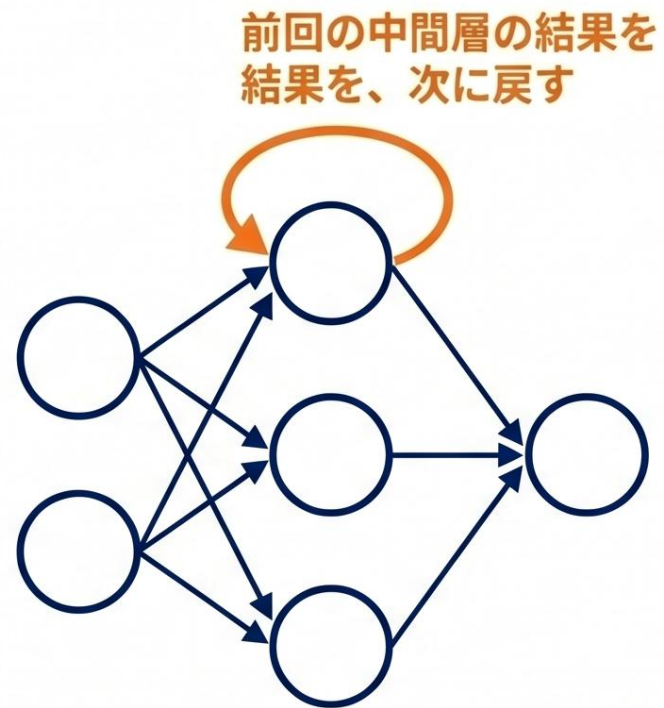
① フィードフォワード型 (通常のニューラルネットワーク)



入力層 中間層 出力層

データは左から右へ一方に
流れる。

② リカレント（再帰）を追加

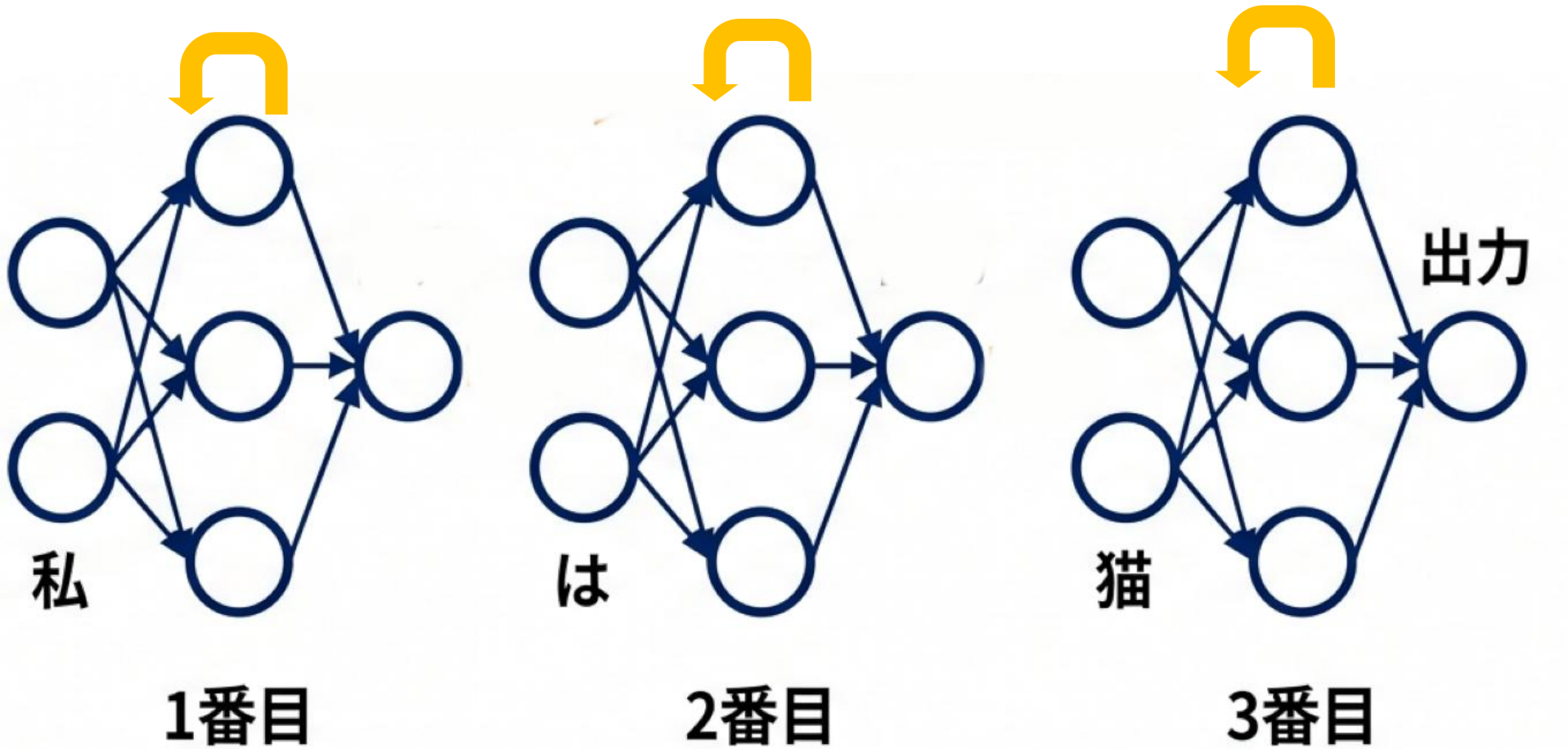


入力層 中間層 出力層

中間層の結果を次の処理に戻す
矢印（再帰）を加える。

ニューラルネットワークの再帰（リカレント）の仕組み

入力：私 は 猫



「私」の記憶が残る

「は」の記憶が残る

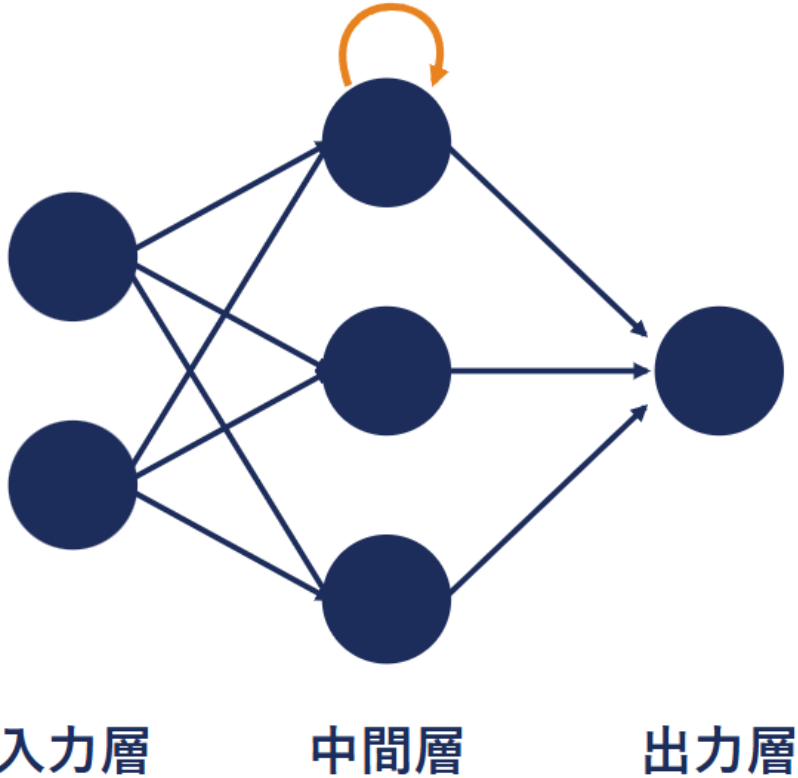
「猫」の記憶が残る

今の入力と記憶で処理を行う 今の入力と記憶で処理を行う

古い情報の長期間記憶を可能にする LSTM

① リカレント（再帰）型

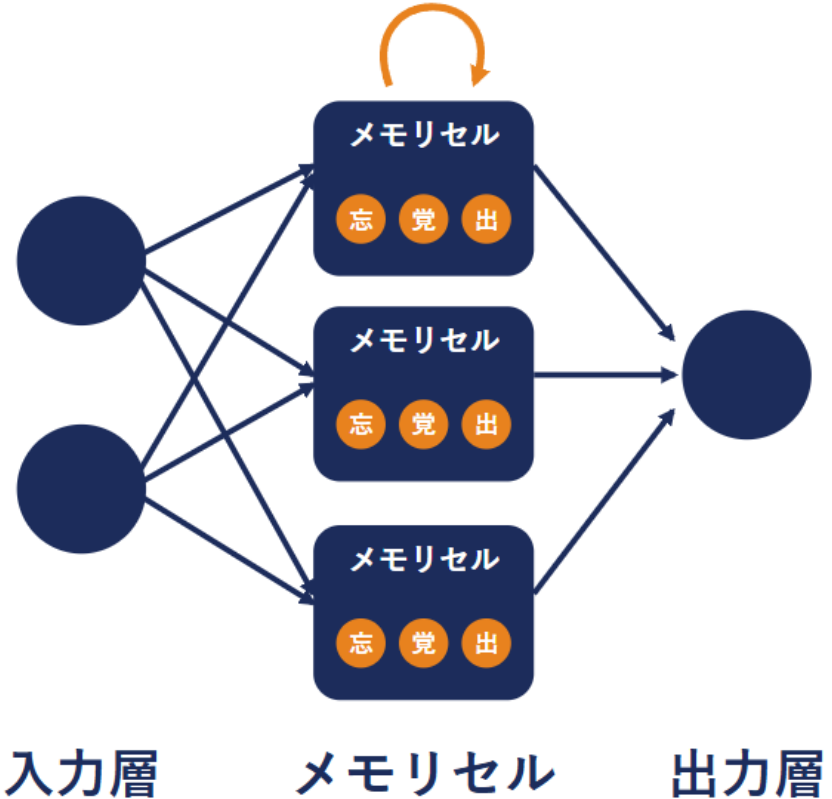
（中間層が結果を覚える）



前回の結果をそのまま次に戻す。
長い記憶は苦手。

② LSTM（記憶セルを追加）

忘れる・覚える・出すを制御



3つのゲートで情報を取捨選択し、
重要な記憶を長く保てる。

再帰（リカレント）と LSTM — 記憶の保ち方の違い

素のRNN



直前の状態だけを伝える。離れた情報は薄れる（勾配消失）

LSTM



得意分野

素のRNNが向く

- 短い時系列（直近の数ステップで答えが出る）
- 軽量・低資源（組み込み・リアルタイム）

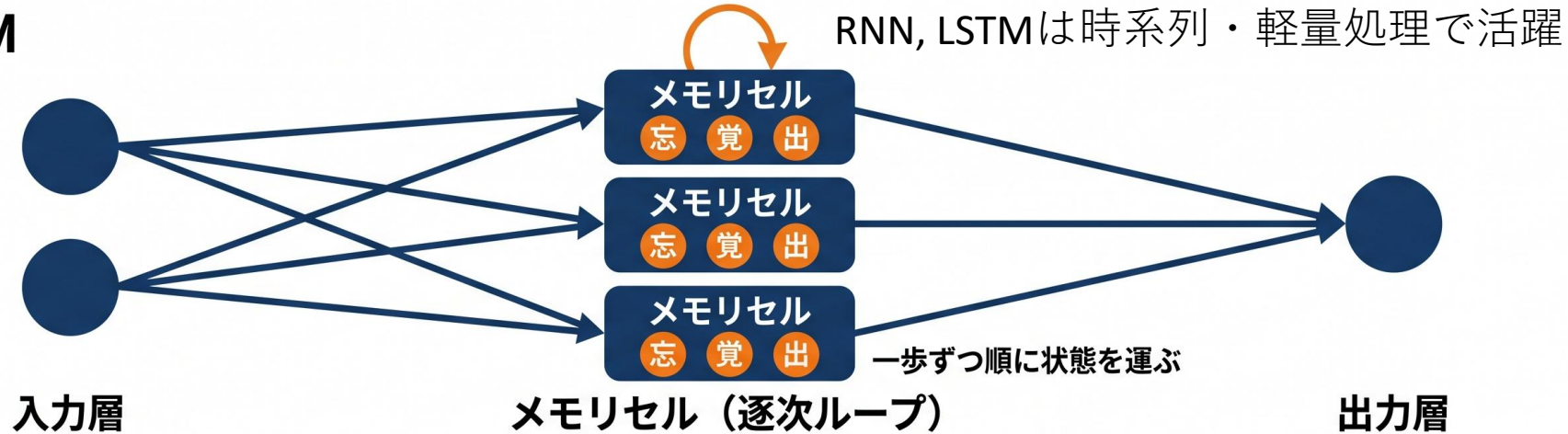
LSTMが向く

- 長い系列・離れた情報の保持が要るタスク（翻訳・文章生成・音声認識）
- 長期依存の時系列予測（株価・需要予測）

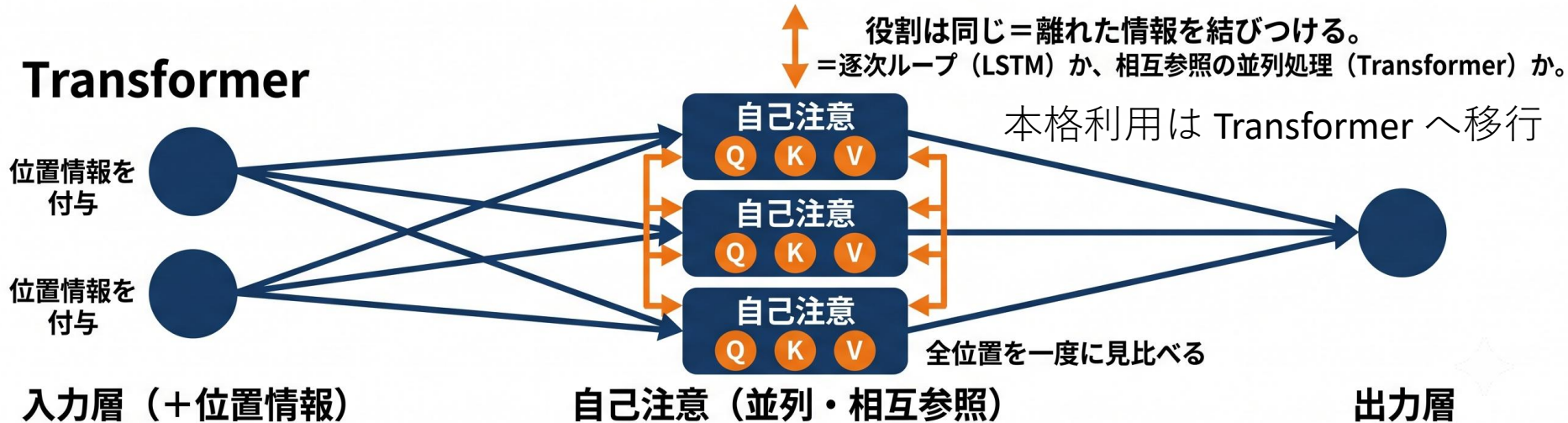
潮流：系列データはTransformerへ
言語・音声を中心に主役はTransformer系へ移行

系列データを扱う AI の主流は Transformerへ

LSTM



Transformer



無関係ではない。Transformerは『逐次で記憶を運ぶ』役割を『全位置を直接見比べる自己注意』で置き換えたもの。

Transformer と文章

離れた語どうしを直接結びつける

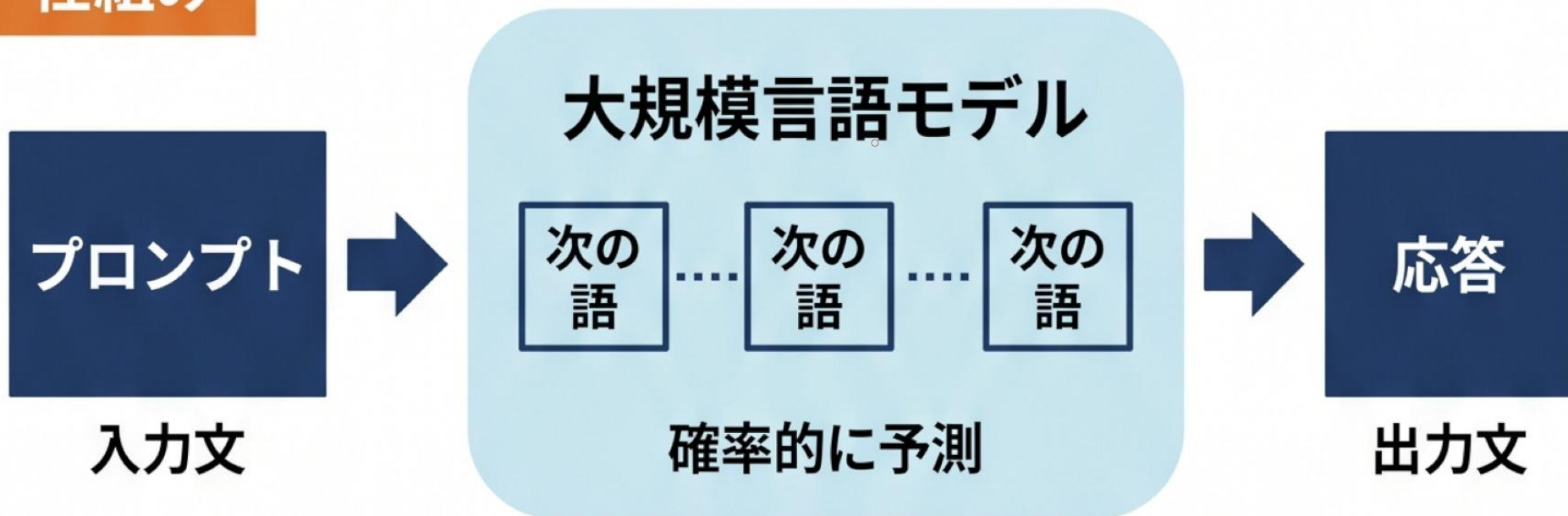
Transformer



その **本** を 昨日 買った
けど まだ **読んでいない**

言葉は離れた位置どうしが関係しあう。
Transformer はこれを一度に見比べられる。

仕組み



検索ではなく、学習した分布から最も自然な語の連なりを生成

対話AIの中身は Transformer



入力文から、**次に来る語を確率的に予測**。それを**繰り返し替え**して、最も自然な語のつながりを生成

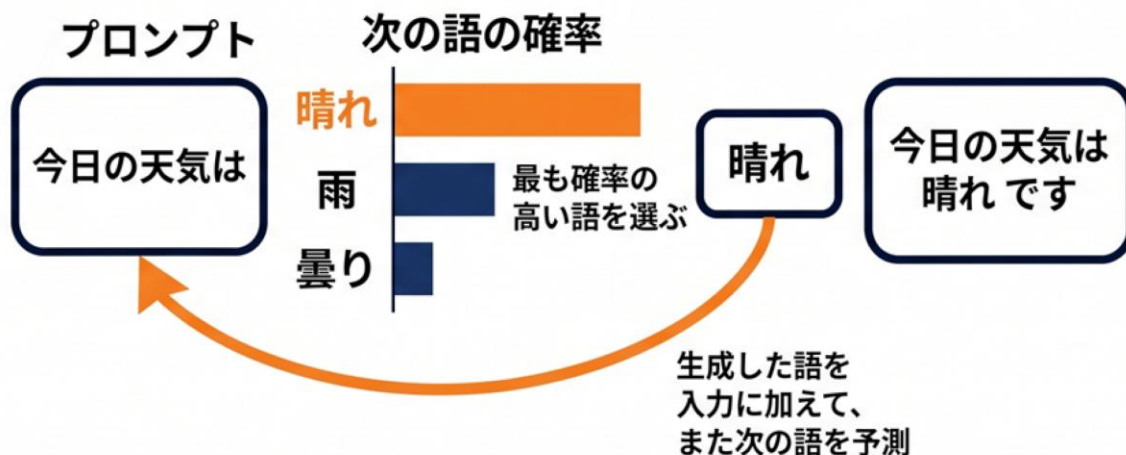
対話AI (ChatGPT など)

LLM (大規模言語モデル)

言葉を確率的に予測する中身

Transformer

予測を支える仕組み



対話AIの生身は Transformer



ChatGPTのような対話AIは、言葉の系列を扱うために
Transformer という仕組みでできている。

身近な対話AI

中身を開けると

なぜ向くのか



離れた語どうしを直接結びつける

その本を昨日買った
けどまだ読んでいない

言葉は離れた位置どうしが関係しあう。
Transformer はこれを一度に見比べられる。

例：ChatGPT などのチャットAI
入力：言葉の系列／出力：言葉の系列



演習

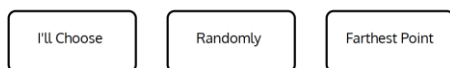
演習 1. クラスタリング学習サイトの活用

- サイトにアクセス:

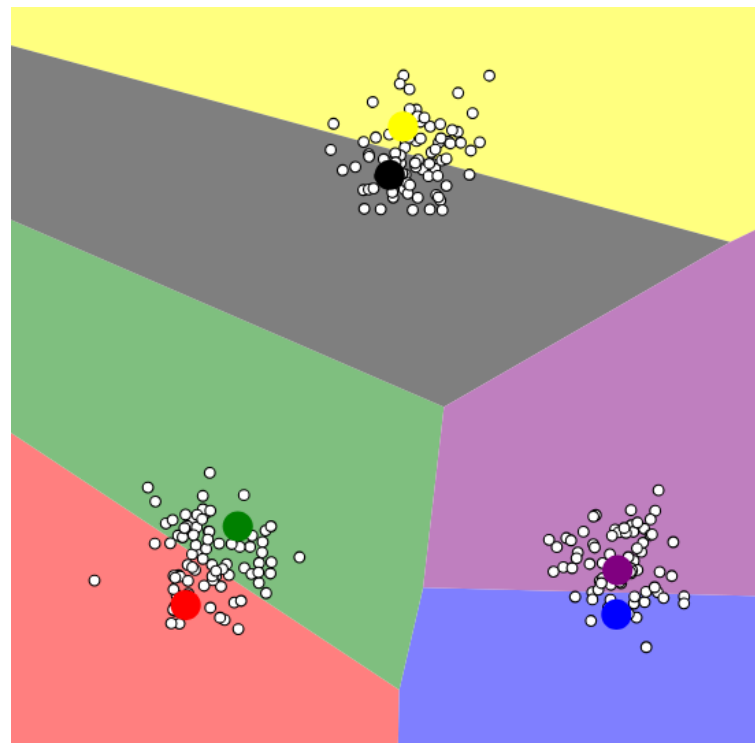
<https://www.naftaliharris.com/blog/visualizing-k-means-clustering/>

- **データセット**: 「Randomly」 を選択

How to pick the initial centroids?



- **分布タイプ**: 好みのものを選択 (例: Gaussian Mixture)
- 「Add Centroid」 をクリック。(自分が望むクラスタ数だけ、クリックする)。結果を確認。収束を観察しながら繰り返す
- やり直す場合は「Restart」 をクリック

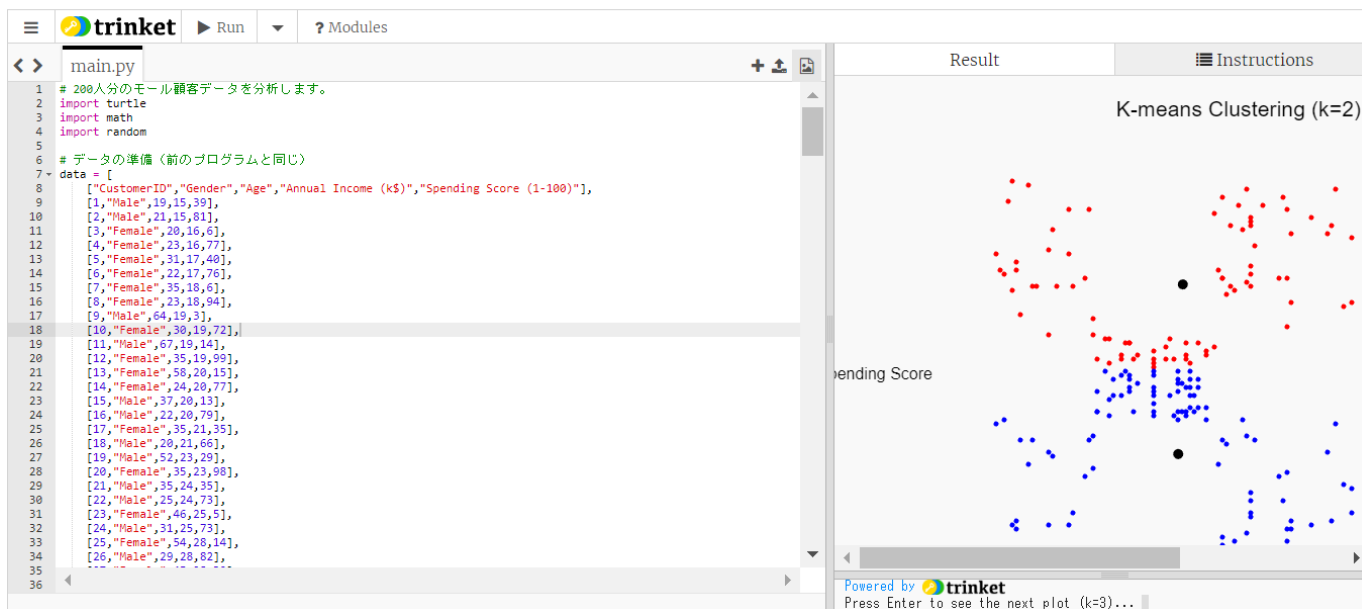


演習 2. クラスタリングをPythonプログラムで行ってみる

① trinket の次のページを開く

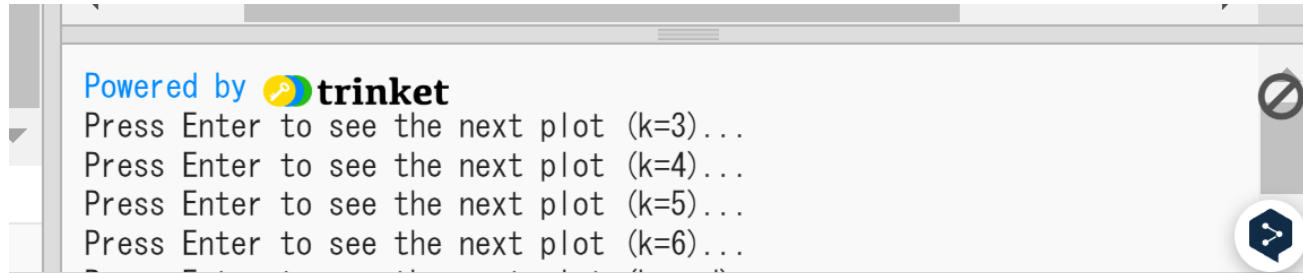
<https://trinket.io/python/93ab2fb15488>

② 実行結果が，次のように表示されることを確認



- 実行が開始しないときは、「**実行ボタン**」で**実行**関係が単純であるため、ディープラーニングとは別の方法でクラスタリングを実行

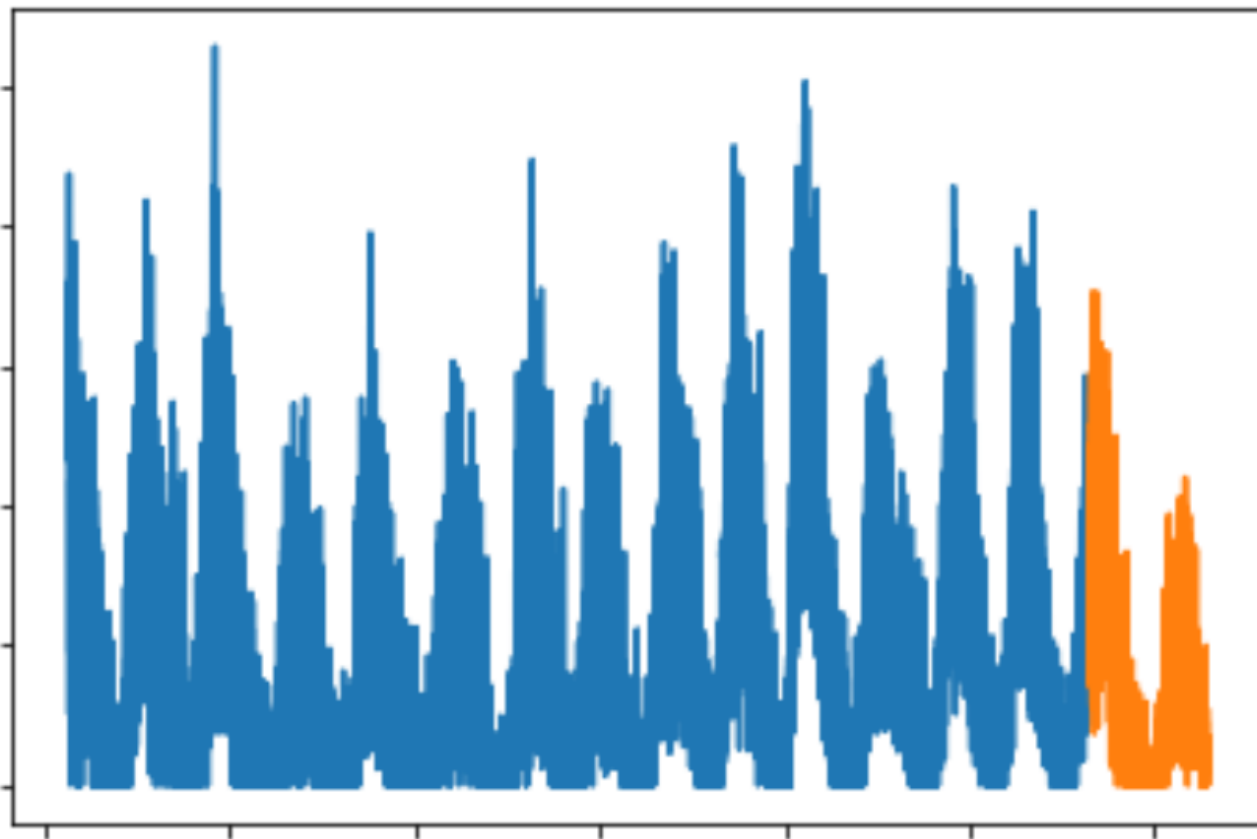
③ 右下の画面（Press Enter to see the next plotと表示されている画面）をクリック。Enter キー



④ **クラスタ数が3に増えるので確認**

その後、③の操作と④の確認を繰り返す

演習 3. LSTM を用いた予測の例



太陽の黒点数の変化

予測

1848年～1999年のデータを用いて、**2000年以降を予測**
(ディープニューラルネットワークによる予測)

系列データの例：太陽黒点観測データ

11314	1848	12	23	1848.977	353	23.8	1
11315	1848	12	24	1848.980	240	19.6	1
11316	1848	12	25	1848.982	275	21.0	1
11317	1848	12	26	1848.985	352	23.8	1
11318	1848	12	27	1848.988	268	20.8	1
11319	1848	12	28	1848.990	285	21.4	1
11320	1848	12	29	1848.993	343	23.5	1
11321	1848	12	30	1848.996	340	23.4	1
11322	1848	12	31	1848.999	238	19.6	1
11323	1849	1	1	1849.001	287	20.9	1
	year	month	day	dec_year	sn_value	sn_error	obs_num
72855	2017	6	21	2017.470	35	1.0	41
72856	2017	6	22	2017.473	24	0.8	39
72857	2017	6	23	2017.475	23	0.9	40
72858	2017	6	24	2017.478	26	2.3	15
72859	2017	6	25	2017.481	17	1.0	18
72860	2017	6	26	2017.484	21	1.1	25
72861	2017	6	27	2017.486	19	1.2	36
72862	2017	6	28	2017.489	17	1.1	22
72863	2017	6	29	2017.492	12	0.5	25
72864	2017	6	30	2017.495	11	0.5	30

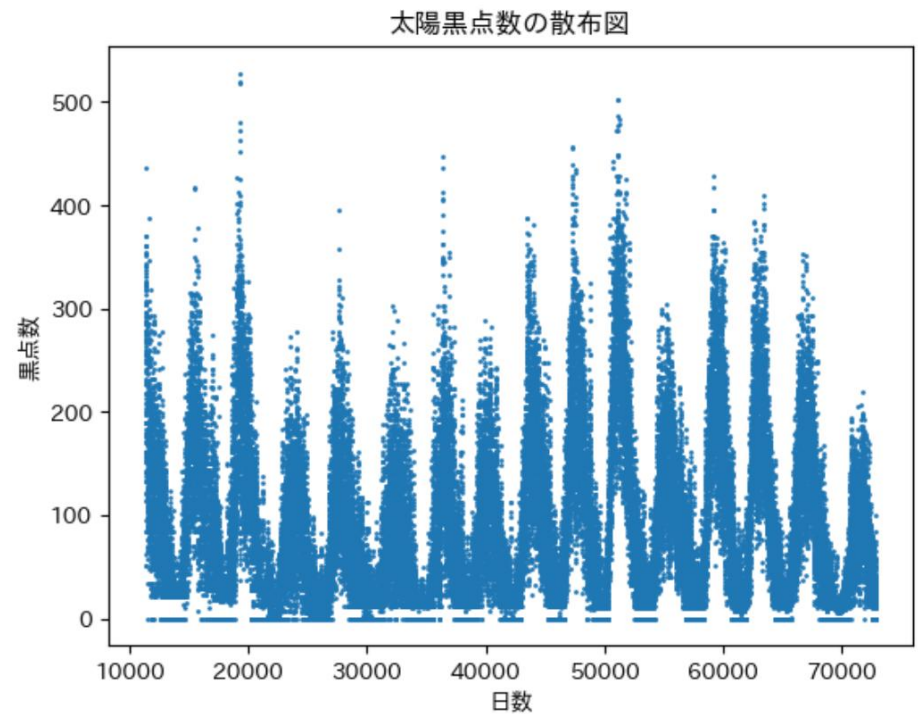


年月日の値 黒点数

黒点数の観測データ

1848年12月23日～2017年6月30日の
毎日の黒点数データが公開されている

https://data.heatonresearch.com/data/t81-558/SN_d_tot_V2.0.csv



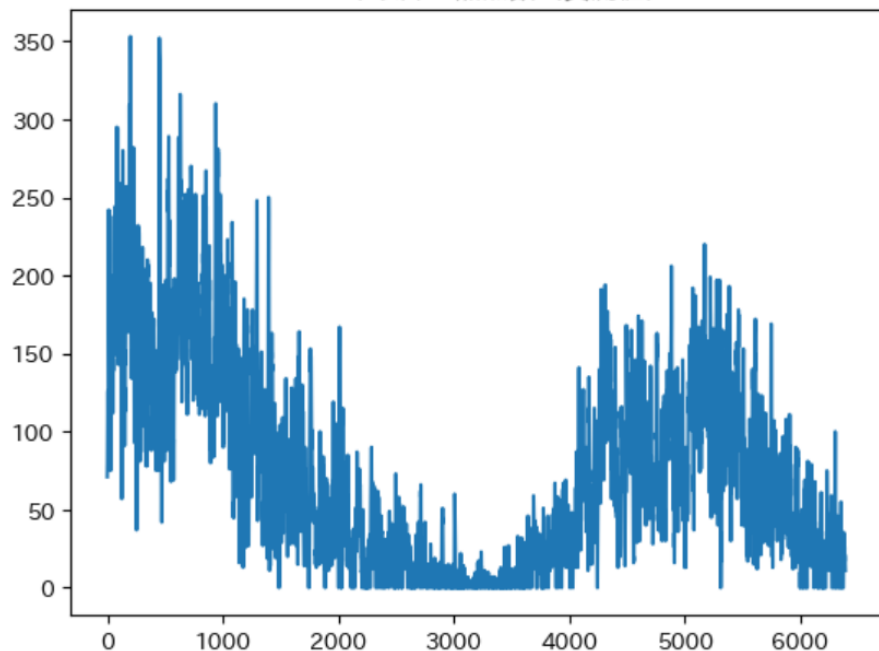
散布図でプロット

① Google Colaboratory のページを開く

https://colab.research.google.com/drive/1tLfijmXCJyg--4Jr5pbzV5G7p_3LILQD?usp=sharing

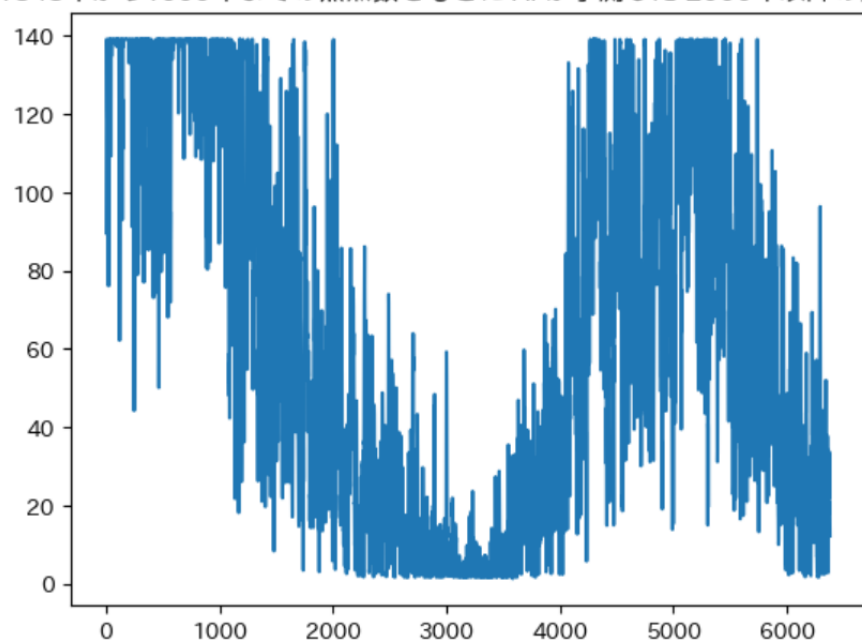
② プログラムと実行結果（左が正解、右がLSTMによる将来予測）を確認

2000年以降の黒点数（実測値）



実際のデータ

1848年から1999年までの黒点数をもとに AI が予測した 2000年以降の黒点数



過去データからの予測