

情報科学 【AI・データサイエンス】

第2回 AIと機械学習

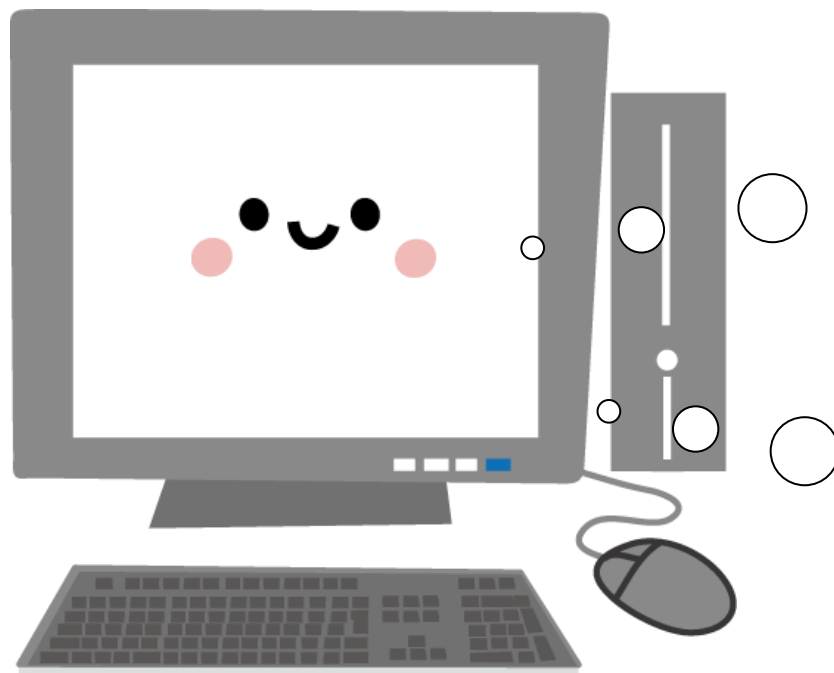
人工知能（AI）とその応用例
機械学習による人工知能の作り方
最近のAIの正体：深層ニューラルネットワーク
いまのAIにできないこと

人工知能（AI）と その応用例

人工知能（AI）とは

人工知能(AI)

- = 人間の知能を真似する機械



冷蔵庫の奥に古い
豚肉を見つけたから、
今晚のおかずに
こっそり混ぜちゃおう

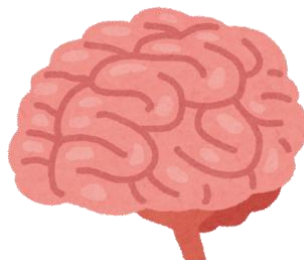
おや、〇子さんの
機嫌がわるいぞ。
きっとテストのヤマが外
れたんだろう

- ところで、**知能**って何？

我々の知能， 具体的にはどんなことを？

- 予測や発見、パターン認識を行う
- 数学の問題を解く
- 文章を読んで理解し、さらに翻訳する
- 会話する
- 将棋のようなゲームをする
- ラーメンをおいしいと思う
- 音楽を聴いて感動する
- ジョークを聞いて笑う
- 作文をする
- 絵を描く
- 恋愛感情を持つ
- 人の考え方に共感する
- 気になるところに注意を向ける

超多機能



1つのAIにこれらすべての機能を持たせるのは難しそう

特化型AIとは？



⇒



- 特定の知能だけを人工的に実現したAI
 - =特定のことはできないAI
- 現在利用されているすべてのAIは、特化型AI
 - 画像認識AI
 - 対話AI (チャットボット)
 - 将棋AI
 - 天気予報AI...
- 天気予報AIに将棋は無理，画像認識AIに対話は無理...

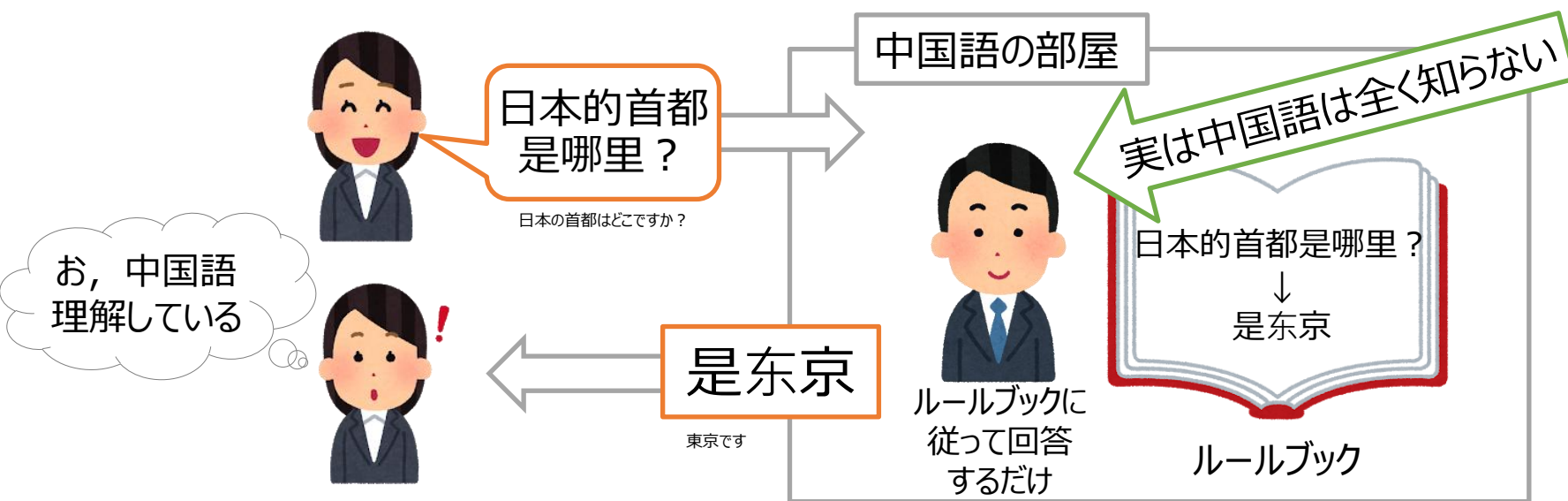
別物として独立して作られたもの



特化型AIは本当に智能なのか？

「中国語の部屋」 [Searle, *Minds, Brains, and Programs*, 1980] (1/2)

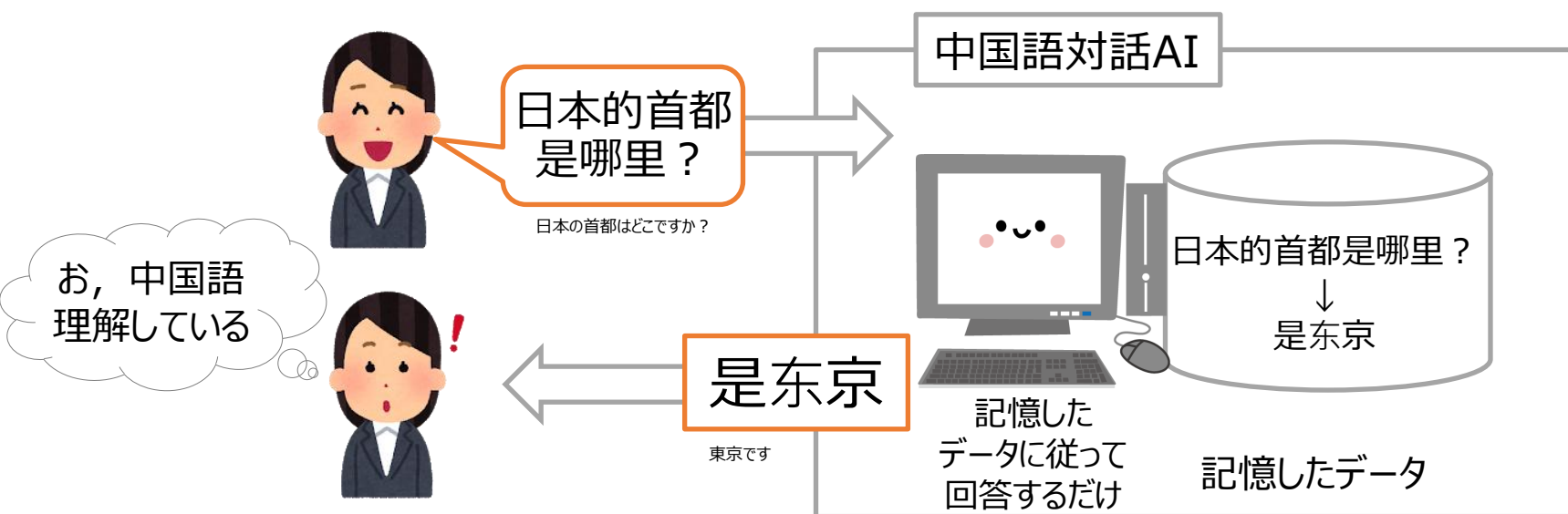
- 部屋の中の人是中国語を全く理解していないのに、外から見ると中国語に関する智能を持ってるように見える



- 「中の人」  は「中国語について何かを考える」ような智能は持っていない

特化型AIは本当に智能なのか？ 「中国語の部屋」[Searle,1980] (2/2)

- 現在の特化型AIが実現しているのは，まさにこの形式



- 前ページと同じ理由で，これを「智能」と呼ぶには抵抗がある
 - →そのため「弱いAI」とも呼ばれる

汎用AIとは？



=



- 我々の知能と同じ柔軟さと多機能性を持つ「強いAI」

- 予測や発見、パターン認識を行う
- 数学の問題を解く
- 文章を読んで理解し、さらに翻訳する
- 会話する
- 将棋のようなゲームをする
- ラーメンをおいしいと思う
- 音楽を聴いて感動する
- ジョークを聞いて笑う
- 作文をする
- 絵を描く
- 恋愛感情を持つ
- 人の考え方に共感する
- 気になるところに注意を向ける

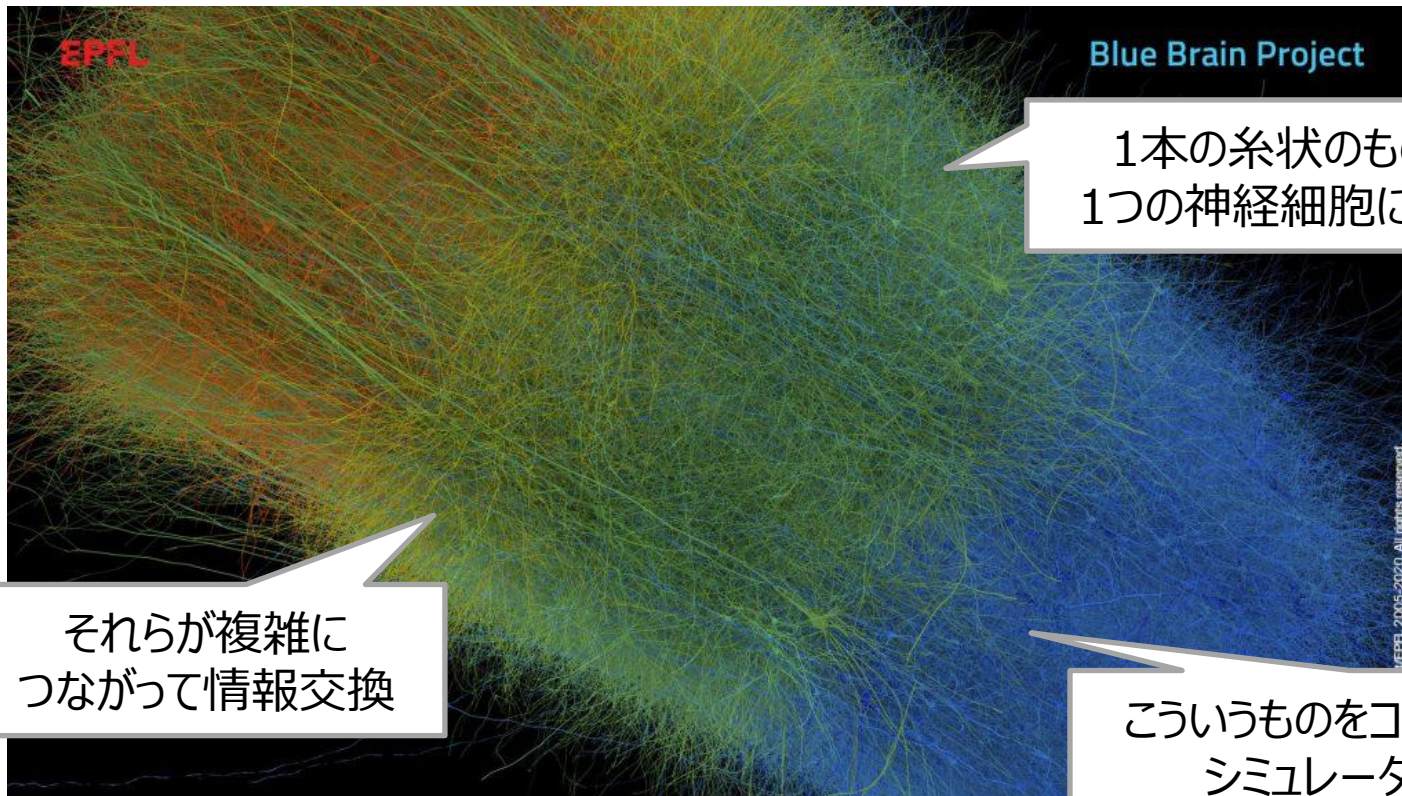


このAI一つで
何でもできる!

- できたらすごいが、現状ではまだ検討段階
 - 「どんなAIが強いAIなのか」も定義できていないのが実情

汎用AIの実現法の一つ：「全脳シミュレーション」

1. 脳を構成する神経細胞(ニューロン) の個々の動きを再現
2. それら神経細胞（ニューロン）を繋げて脳全体を再現



全脳シミュレーションの草分け, EPFL/Blue Brain Project
<https://www.epfl.ch/research/domains/bluebrain/gallery/>

こういうものをコンピュータによる
シミュレータとして実現

あなたのそばにも人工知能

すでに使われている・使われつつある人工知能

身の回りで活用される人工知能： 対話生成AI: ChatGPT, Copilot, Geminiなど

- 「AIの原理を4文ぐらいで説明してください」というプロンプト（質問）に対するテキスト生成結果



AI（人工知能）は、大量のデータを学習してパターンや規則を見つけ出し、その知識をもとに推論や判断を行います。これには、ニューラルネットワークや機械学習アルゴリズムが使われ、特に深層学習が近年注目されています。AIは訓練データを用いてモデルを構築し、新しいデータに対しても予測や分類を行うことができます。また、反復的な学習プロセスを通じて、自己改善を行い精度を高めていきます。

ChatGPT (OpenAI社, ChatGPT 4o, 2024年8月生成)

身の回りで活用される人工知能： プログラムの自動生成

●「勇者が姫を助けるロールプレイングゲームをpythonで作って」

(python=プログラミング言語の一つ)

```
python
import random

# キャラクタークラスの定義
class Character:
    def __init__(self, name, hp, attack):
        self.name = name
        self.hp = hp
        self.attack = attack

    def is_alive(self):
        return self.hp > 0

    def take_damage(self, damage):
        self.hp -= damage
        print(f"{self.name}は{damage}のダメージを受けた！ 残りHP: {self.hp}")

    def attack_enemy(self, enemy):
        damage = random.randint(1, self.attack)
        print(f"{self.name}が{enemy.name}を攻撃し、{damage}のダメージを与えた！")
        enemy.take_damage(damage)

# 勇者、姫、魔王の設定
hero = Character("勇者", 100, 20)
princess = Character("姫", 50, 10) # 姫は戦わないので、攻撃力は低め
demon_king = Character("魔王", 80, 15)

# ゲームのストーリー
print("勇者が姫を助けに向かいます！")

# 勇者と魔王の戦い
while hero.is_alive() and demon_king.is_alive():
    hero.attack_enemy(demon_king)
    if demon_king.is_alive():
        demon_king.attack_enemy(hero)
    print()

# 結果の表示
if hero.is_alive():
    print("勇者は魔王を倒し、姫を助け出した！")
else:
    print("勇者は魔王に敗れてしまった...姫は救われなかった。")
```

自動生成された
プログラム

実際に
動かしてみる※

勇者が姫を助けに向かいます！
勇者が魔王を攻撃し、5のダメージを与えた！
魔王は5のダメージを受けた！ 残りHP: 75
魔王が勇者を攻撃し、7のダメージを与えた！
勇者は7のダメージを受けた！ 残りHP: 93

勇者が魔王を攻撃し、4のダメージを与えた！
魔王は4のダメージを受けた！ 残りHP: 71
魔王が勇者を攻撃し、1のダメージを与えた！
勇者は1のダメージを受けた！ 残りHP: 92

勇者が魔王を攻撃し、13のダメージを与えた！
魔王は13のダメージを受けた！ 残りHP: 58
魔王が勇者を攻撃し、6のダメージを与えた！
勇者は6のダメージを受けた！ 残りHP: 86

(中略)

勇者が魔王を攻撃し、18のダメージを与えた！
魔王は18のダメージを受けた！ 残りHP: 4
魔王が勇者を攻撃し、10のダメージを与えた！
勇者は10のダメージを受けた！ 残りHP: 52

勇者が魔王を攻撃し、14のダメージを与えた！
魔王は14のダメージを受けた！ 残りHP: -10

勇者は魔王を倒し、姫を助け出した！

※左のコードをGoogle Colaboratory にcopy&pasteするだけ動かせる

身の回りで活用される人工知能： 画像生成の例

- “Make our society happy with data science”（我々の社会をデータサイエンスで幸せに！）というプロンプトに対する画像生成結果



StableDiffusionWeb.com



ChatGPT (OpenAI社, ChatGPT 4o)

余談：様々なことができるChatGPTでも、特化型AIなのか？

- まあ「まだ」そうですね。今後どうなっていくか見守りましょう...

python

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# 1) データ読み込み (列例: Player, Team, AB, H, 2B, 3B, HR, BB, IBB, HBP, SF, OBP, SLG, ISO)
bat = pd.read_csv('mlb_2024_batting.csv')

# 2) 2024年のwOBA重み (Fangraphs公表値を例示)
w = dict(uBB=0.693, HBP=0.722, 1B=0.878, 2B=1.242, 3B=1.569, HR=2.007)

# 3) 単打列を作成
bat['1B'] = bat['H'] - bat['2B'] - bat['3B'] - bat['HR']

# 4) wOBA計算
num = (w['uBB']*(bat['BB']-bat['IBB'])
      + w['HBP']*bat['HBP']
      + w['1B']*bat['1B']
      + w['2B']*bat['2B']
      + w['3B']*bat['3B']
      + w['HR']*bat['HR'])
den = bat['AB'] + bat['BB'] - bat['IBB'] + bat['SF'] + bat['HBP']
bat['wOBA'] = num / den

# 5) 簡易 wRC+ (球場補正なしのラフ版)
lg_wOBA = bat['wOBA'].mean()
wOBA_scale = 1 / (lg_wOBA - 0.320) # 0.320 は便宜的な常数
bat['wRC+'] = 100 + (bat['wOBA']-lg_wOBA)*wOBA_scale*100

# 6) 上位10打者表示
print(bat[['Player', 'Team', 'wOBA', 'wRC+']]
      .sort_values('wRC+', ascending=False).head(10))

# 7) レート指標の相関行列とヒートマップ
cols = ['wOBA', 'OBP', 'SLG', 'ISO']
corr = bat[cols].corr()
print(corr.round(2))

plt.imshow(corr, cmap='coolwarm', vmin=-1, vmax=1)
plt.xticks(range(len(cols)), cols); plt.yticks(range(len(cols)), cols)
plt.colorbar(label='Correlation')
plt.title('Rate-stat Correlation (2024)')
plt.tight_layout(); plt.show()
```

MLBデータ分析
プログラム

2025.5.1実施
ChatGPT o3

「▲飛車の捨て駒で玉を横に引きずり出し、頭金で仕留める」という基本手筋を覚えるのがねらいです。

diff

```
9 8 7 6 5 4 3 2 1
+-----+
1 | . . . . . 玉 |
2 | . . . . . 歩 |
3 | . . . . . 歩 |
4 | . . . . . 歩 |
5 | . . . . . 歩 |
6 | . . . . . 歩 |
7 | . . . . . 歩 |
8 | . . . . . 歩 |
9 | . . . . . 歩 |
+-----+
先手 (▲) の持ち駒: 飛 金
後手 (△) の持ち駒: なし
手番: ▲先手
```

詰将棋も作れるがまだ怪しい...

- △玉（後手玉）は 2-1 にいます。
- △の歩が 1-2・2-2 を塞いでいるので、玉は初手でそこへ逃げられません。

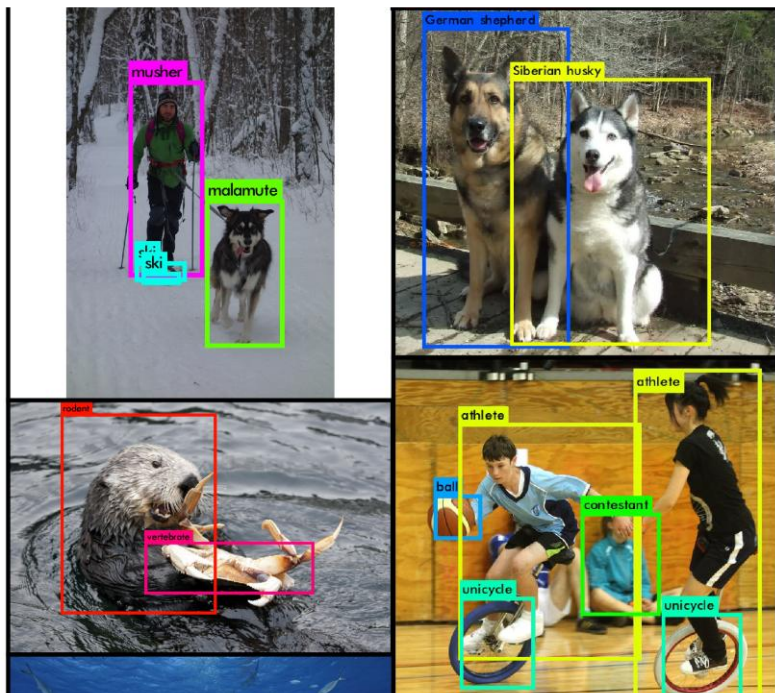
詰手順（▲が攻め方）

手数	▲攻め方	△受け方	解説ポイント
1 手目	▲3-飛打（捨て駒で王手）	-	横一線に飛車を打って王手。▲飛は取られることを前提にした「いけにえ」です。
1 手目の応手	-	△同 玉（飛車を取るしかない）	歩が 1-2・2-2 を塞いでいるため、玉は 3-1 で飛車を取る以外に逃げ道がありません。
3 手目	▲3-二金打（頭金）	読み	金を玉の真上に叩き込みます。 金は 1-1・2-1・4-1・2-2・4-2 をすべてカバーし、玉はどこにも逃げられません。
3 手（▲→△→▲）で読みとなります。			

2025.5.1実施
ChatGPT o3

身の回りで活用される人工知能： 画像認識

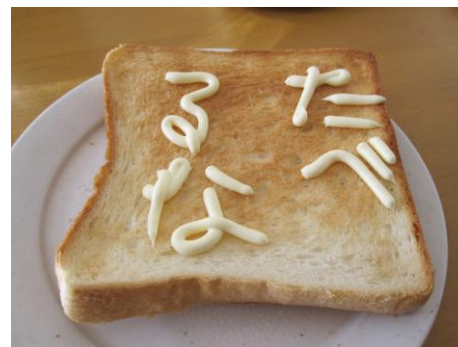
● 基本的な画像認識



[Redmon&Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR2017]

画像中の物体がなんであるかを
自動判定

● 最近の画像認識



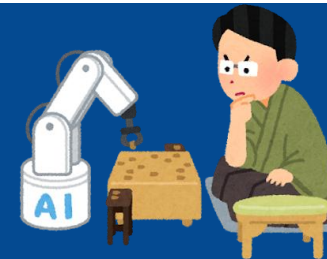
プロンプト：
これはなんですか？



こんがり焼いた食パンの上に、マヨネーズで文字（カタカナらしき形）が描かれているトーストです。パンは白い皿に載せられ、木目のテーブルの上に置かれています。

ちょっと間違ってるが(カタカナ→ひらがな)
詳細な説明も可能

ゲームと人工知能： コンピュータ将棋



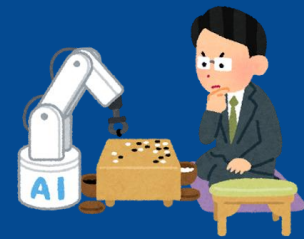
- 第2期電王戦 2017年4～5月
 - 第2期叡王・佐藤天彦にコンピュータが2連勝！



<http://denou.jp/2017/>

- これで電王戦は終了
 - 「人間とコンピュータが同じルールで真剣勝負をするという歴史的役割は終わった」

ゲームと人工知能： AlphaGo (囲碁)



- 2016年3月に韓国のプロ棋士に4勝1敗と勝ち越し
- 深層学習 + 膨大な棋譜データ + 計算機どうしで戦う訓練



<http://www.asahi.com>

- 2017年5月に中国のトッププロにも勝利→プロジェクト中止へ

入試に挑むAI：“東ロボくん”： 「ロボットは東大に入れるか」



- 2013.11.28
 - 「センター試験」模試で、私大579校のうち、403大学の合格可能性がA判定！
- 2015.11.14
 - 全大学の6割にあたる474大学1094学部で合格の可能性が80%以上
 - 偏差値57.8

東口ボクんの模試成績（カッコ内は昨年の成績）

大学入試センター試験の模試（マーク式）			
	得点	全国平均	偏差値
英語（筆記）	80(95)	86	48.4(50.5)
英語（リスニング）	16(－)	24.6	40.5(－)
国語（現代文＋古文）	90(69)	105.4	45.1(54.2)
数学ⅠA	75(40)	45.5	64(46.9)
数学ⅡB	77(55)	42.8	65.8(51.9)
世界史B	76(52)	45.9	66.5(56.1)
日本史B	55(44)	46.6	54.8(48.2)
物理	42(31)	49.4	46.5(49.0)
合計 ＝950点満点	511 (386)	416.4	57.8(47.3)
2次試験の模試（論述式）			
	得点	全国平均	偏差値
地理歴史 ＝60点満点	21	17.2	54.1
数学（文系） ＝80点満点	39	25	59.2
数学（理系） ＝120点満点	20	31.1	44.3

「昨年は英語（リスニング）を受験せず、9000点満点だった。国語の漢文は受験していないが、平均点には漢文も含まれている。」

■ 東ロボくんの模試成績（カッコ内は昨年の成績）

入試に挑むAI： 最近は恐ろしいことに...



- 2025年，LifePrompt社が最新AIに東大入試を解かせたら...
 - 「解かせた」といっても，試験問題用紙を画像として入力しただけ... それ以上のインストラクションはなかったにも関わらず...

共通テスト(2025)：ChatGPT o1・DeekSeek R1  LifePrompt

科目	ChatGPT o1	DeekSeek R1
英語リーディング	98 (98%)	88 (88%)
英語リスニング	93 (93%)	73 (73%)
国語	188 (94%)	178 (89%)
数学1A	80 (80%)	59 (59%)
数学2BC	93 (93%)	85 (85%)
世界史探究	97 (97%)	97 (97%)
日本史探究	79 (79%)	70 (70%)
物理	84 (84%)	67 (67%)
化学	79 (79%)	75 (75%)
理科基礎 (地学・生物)	93 (93%)	88 (88%)
情報	92 (92%)	78 (78%)
文系合計	913 (91%)	816 (82%)
理系合計	904 (90%)	800 (80%)

※ChatGPT o1・DeekSeek R1いずれも、プロンプトなしで画像を読み込ませて実行。

東大(2025)：ChatGPT o1・DeekSeek R1  LifePrompt

採点協力：河合塾 / 答案作成：LifePrompt (ChatGPT o1・DeekSeek R1いずれも、プロンプトなしで画像を読み込ませて実行)

科目	ChatGPT o1	DeekSeek R1	合格最低点
文科1類	379	351	336
文科2類			332
文科3類			321
理科1類	374	共通テストで 足切り ^(※2)	321
理科2類			313
理科3類		369	368

※1 本実験では、地理3科目を解かせ、成績の良かった地理・世界史の点数を採用。なお、日本史の結果を採用した場合でも文科の3科類には合格水準だった。

※2 2025年度東京大学の、理科1類・2類はそれぞれ、808点・814点が足切りラインだったのに対し、DeekSeek R1の共通テストは800点だった。

科目別分析などの詳細は↓

<https://note.com/lifeprompt/n/n0078de2ef36b>

AI等を活用したビジネス

シェアリングエコノミー

様々なもののシェア（賃借や売買や提供）に基づく経済活動

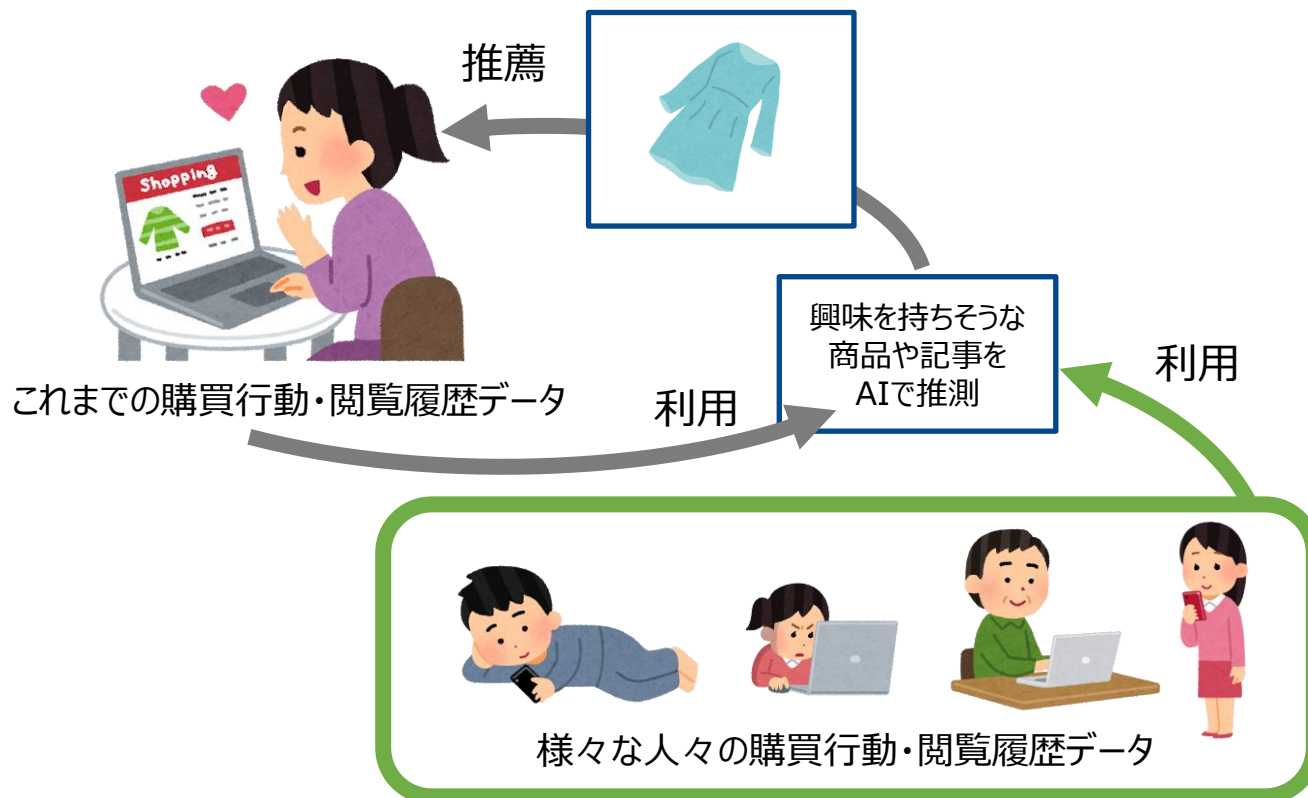
- シェア対象：場所，乗り物，モノ，人，など
 - 場所：宿泊施設(Airbnb) やオフィスの貸し
 - 乗り物：自動車（Uber ではドライバーを含めて共有）
 - モノ：ネットフリーマーケット
 - 人：家事代行者，配達人



- Aiがどう利用されている？
 - いつ・どこで・どのような共有依頼が来るかの予測
 - シェアすべきリソース（部屋や車）の最適配分
 - 需要・供給に応じた利用価格の自動決定
(ダイナミックプライシング)

商品の推薦 (リコmend)

- AIが「このような商品を過去に買ったり, このようなweb ページを過去に閲覧したり, このようなメッセージを送ったりする人は, このような商品を買う可能性が高い」を推測



顧客監視（サーベイランス）

- カメラを店舗や街中に設置して，ヒトの動きやモノの動きを観察するサービス

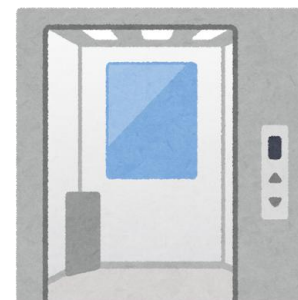


- 防犯

- 万引きなど犯罪の抑制

- 異常検知

- 例えば，人間が何千台ものエレベータ内の状況を同時監視するのは不可能
 - 「普段起こっていない状況」となったエレベータをAIが判断 → 人間に通報



- 顧客の行動把握

- 「どのぐらいの年齢の人が，どの商品棚の前に，どれぐらい滞在するか」をAIで把握(トライアル社)
 - 無人店舗応用(Amazon Go)



Wikipedia "Amazon Go"



<https://www.trial-net.co.jp>

カメラが天井に

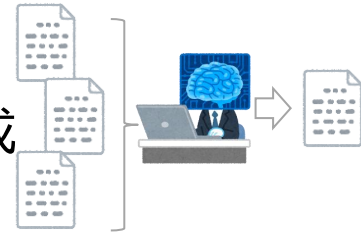
- 河川氾濫や渋滞監視等にも利用

デジタルトランスフォーメーション(DX)

- AI を含む様々な情報処理技術によりビジネスや様々な組織の効率を向上させること



- DXの例：ロボティックプロセスオートメーション(RPA)
 - 複数のファイルを解析・統合し，作業に必要なファイルを自動生成
- DXの例：名刺管理
 - 社内に集まった顧客の名刺を管理，ある商談に適切な顧客を自動抽出



- 今後DXが多用されると考えられているオフィス業務
 - 意思決定，予測，人事，経理ミス発見，業績評価など



機械学習による 人工知能の作り方

機械学習とは

コンピュータ（AI）も学習なしでは、賢くならない

みなさんの学習



たくさんの
練習問題を解いて
正しく回答できる
ように準備

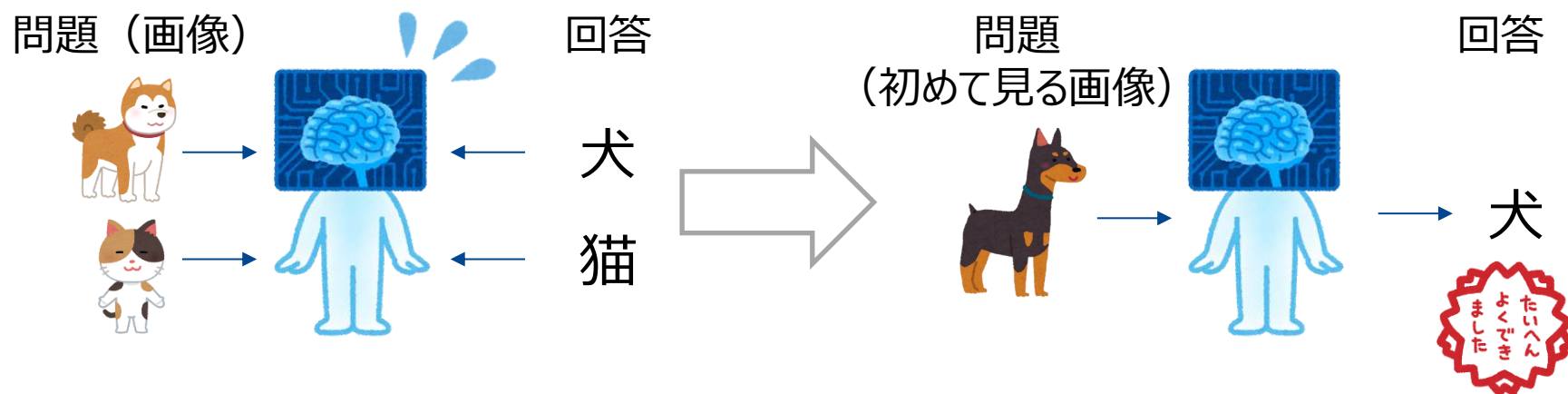


初めて見る
テストの問題でも
正しく回答できる！



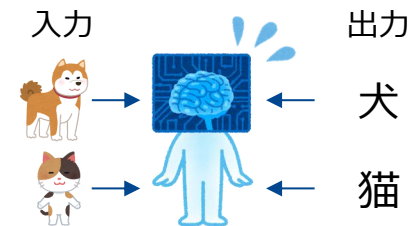
AIの学習も似ている

- 画像認識AIの場合

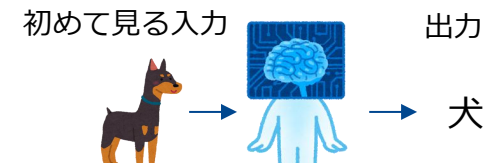


機械学習 = (ザックリ言えば) AIの学習のこと

- AIも、最初からカシコイわけではない
- 「こういう入力には、こう出力しなさい」といった「例」をたくさん用意し、その通り出力するように、AIを調整する必要
 - これを（教師あり）**機械学習**と呼ぶ

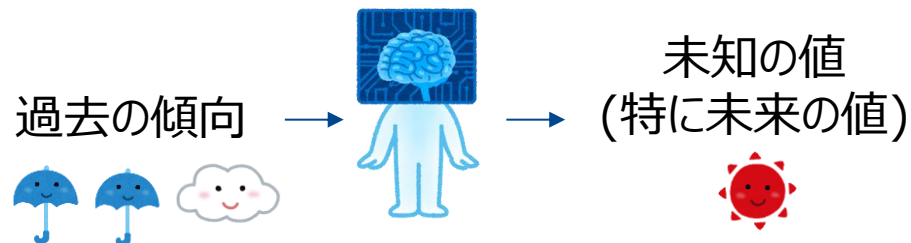


- 十分な例でAIを学習できれば、「（学習に使ってない）初めて見る入力にも、正しい出力を返すことができる」
 - これが嬉しいところ！

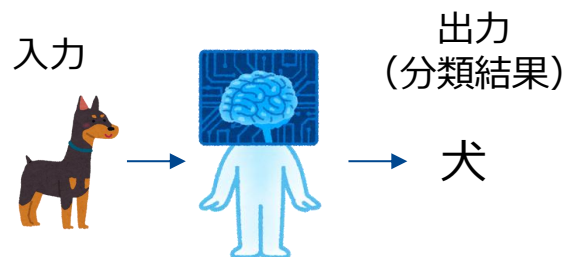


機械学習によって何が可能になるか？ その代表例

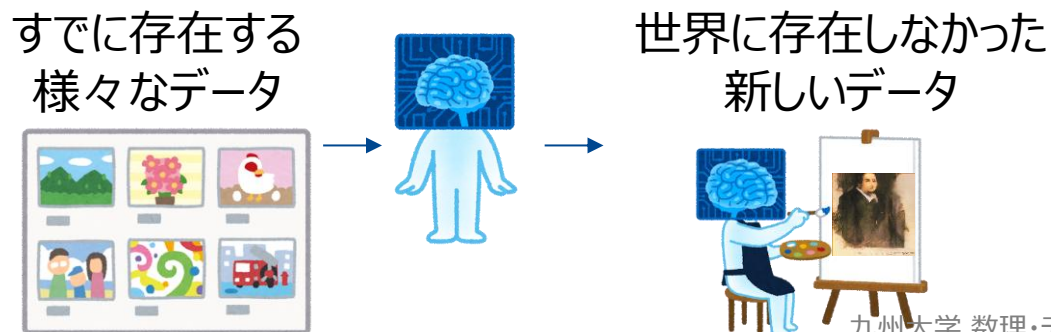
● 予測



● 認識・分類

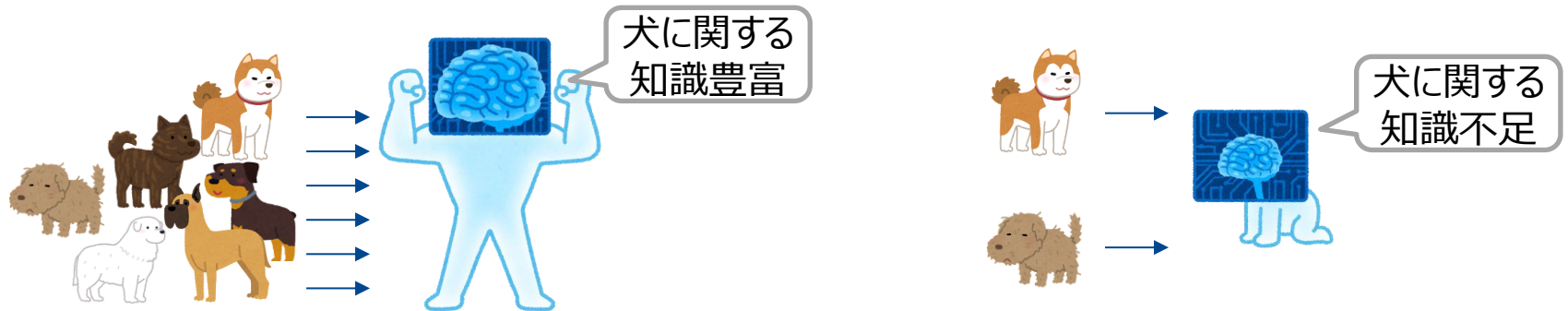


● 生成

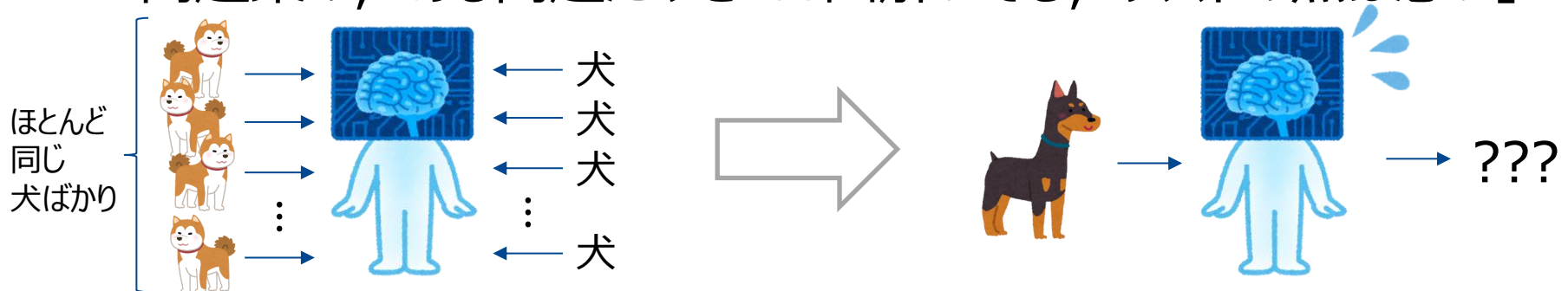


機械学習とデータ

- データが十分にないと，学習しても性能が出せない
 - 「問題集を1,2ページ解いた程度では，テストの点は悪い」のと同じ



- データがたくさんあっても，似たようなものばかりではダメ
 - 「問題集の，ある問題だけを100回解いても，テストの点は悪い」

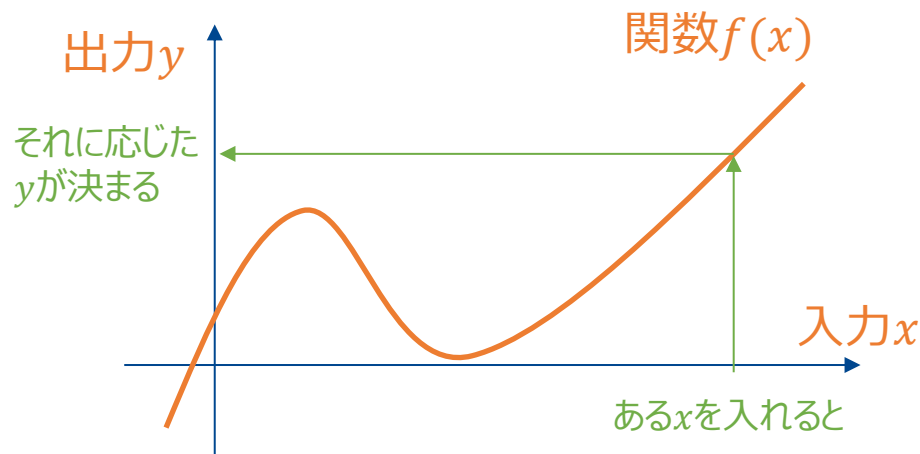


AI・機械学習について もうちょっとだけ詳しく

苦手な人のためのヒント：
「関数」＝「なんか入れたら，なんか出てくるもの」です

突然ですが、関数って習ったことがありますよね...

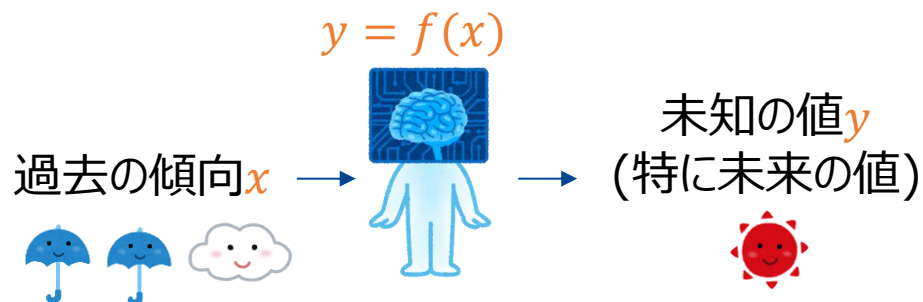
- 教科書で習った関数はこんな感じ？



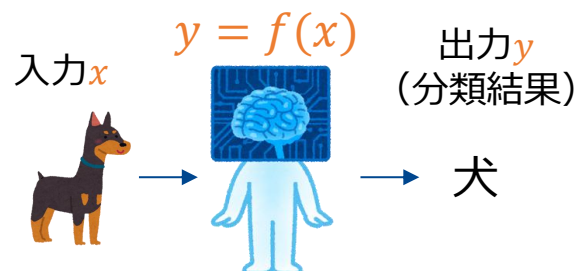
- 要は「 x を与えたら(何かしらのルールに従って) y を返すもの」
 - 難しく考えなくてOK
 - カラオケ採点も関数 (歌を与えたら, 点数を返す)
 - お笑いを見る人間も関数 (お笑い動画を与えたら, 笑顔を返す)

実は, AIとは関数 $y = f(x)$ である!

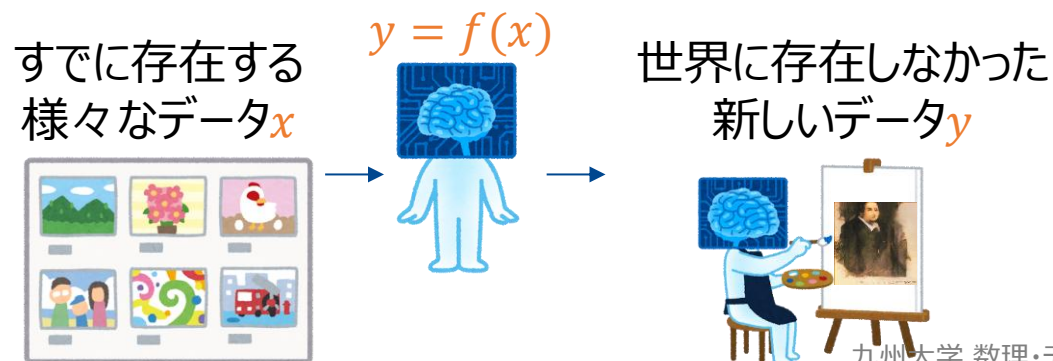
● 予測



● 認識・分類

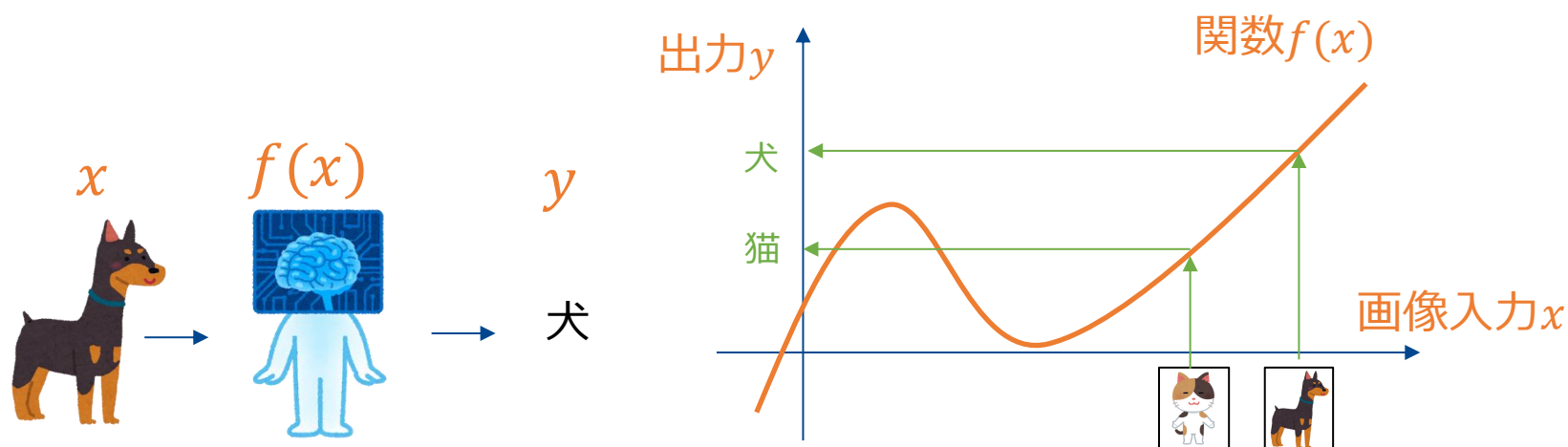


● 生成



無理やり「それ」っぽい

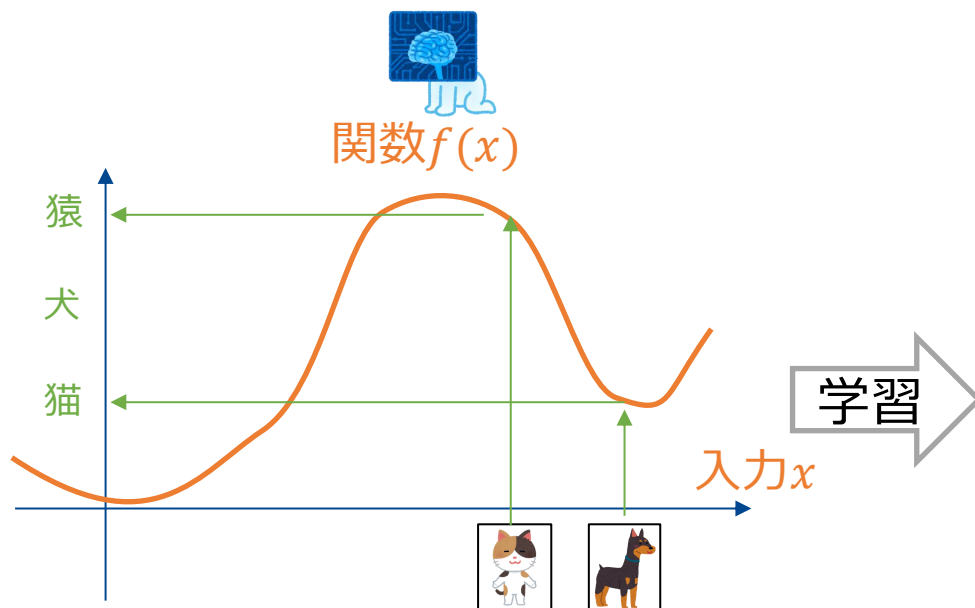
画像認識AIを関数の絵にしてみると...



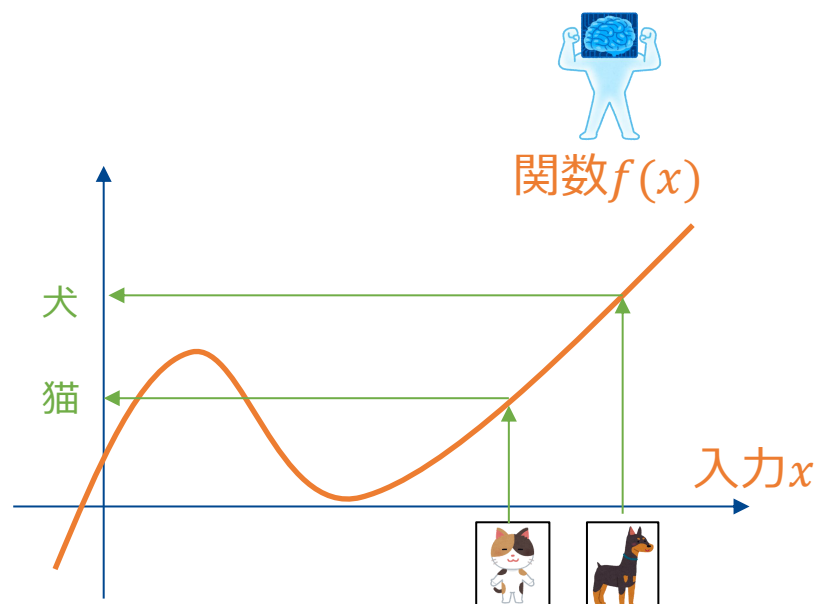
注：非常に大雑把な絵で，不正確な部分もありますが...

従って、機械学習とは、データによって、関数 $y = f(x)$ を望ましい形に「いじる」方法

- 画像認識ならば…



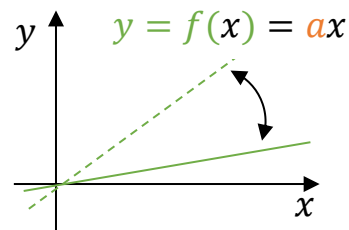
学習前は全然間違った
答えを出す



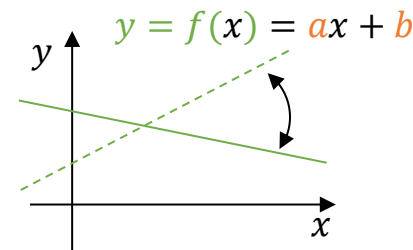
学習により、正しい答えを
出せる関数となる

関数の形をいじる？ → 「パラメータ」をいじる

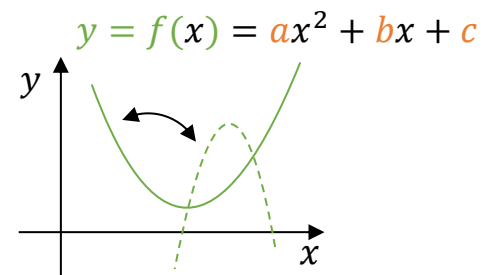
- 関数 $y = ax$ は, a をいじると傾きが変わる



- 関数 $y = ax + b$ は, a や b をいじると傾きや高さが変わる

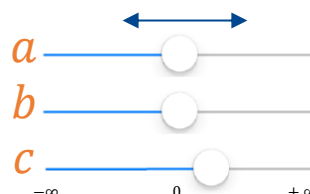


- 関数 $y = ax^2 + bx + c$ は, a や b や c をいじると色々変わる

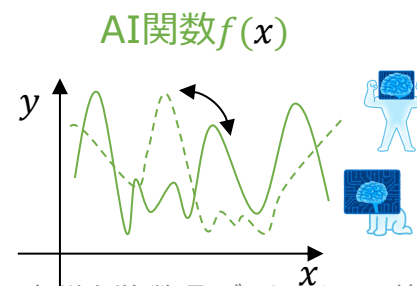


- これら, a や b や c をパラメータと呼ぶ

- 「調節スライダー」のようなもの

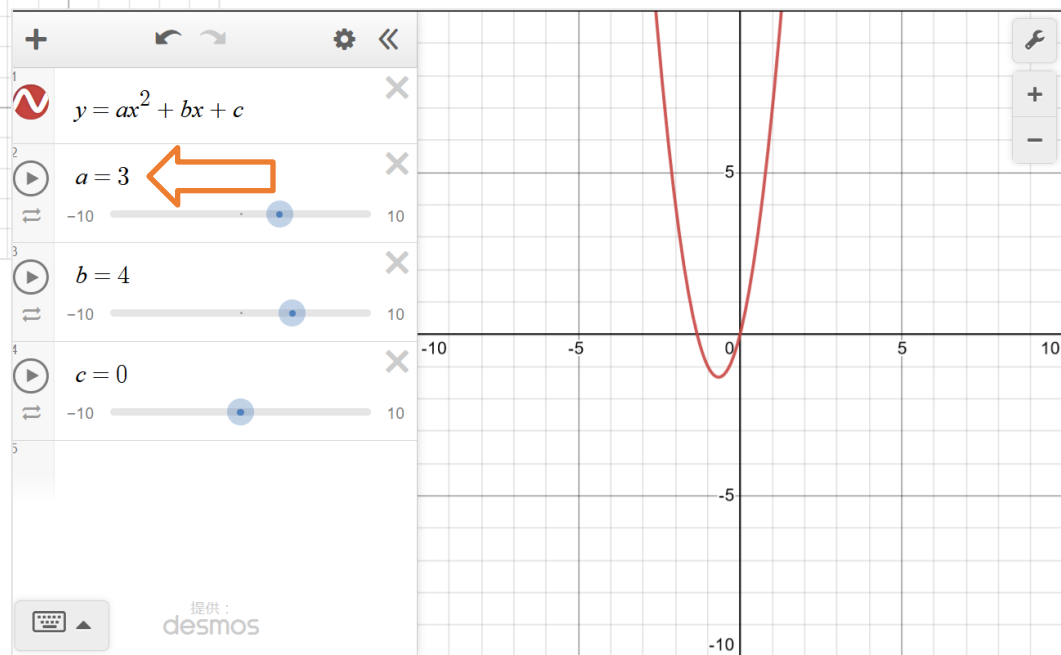
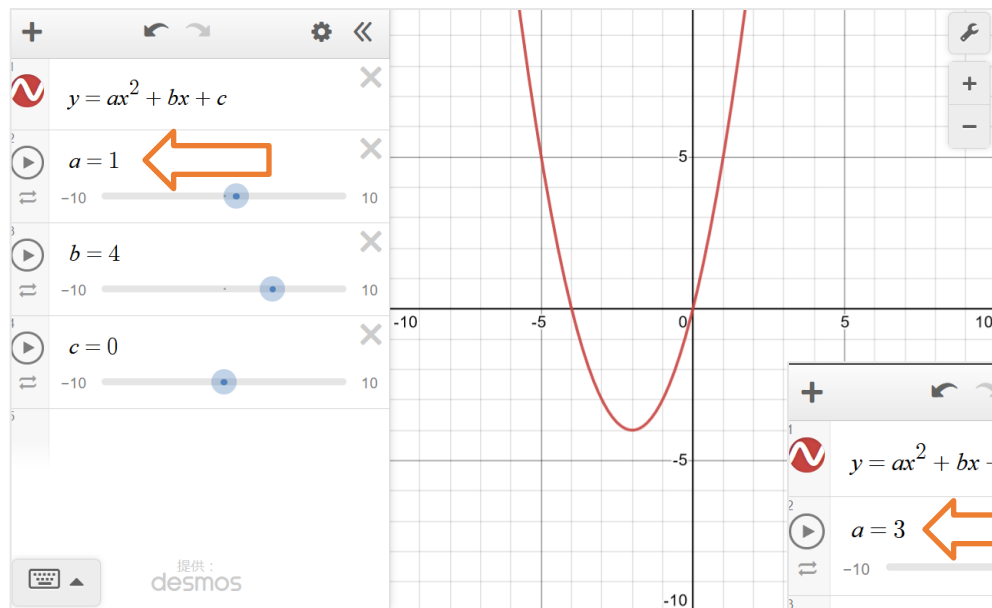


- AI用の関数 $y = f(x)$ にもパラメータがあって, それをいじると形がいろいろ変わる (= 性能も変わる)



参考：パラメータをいじってみよう

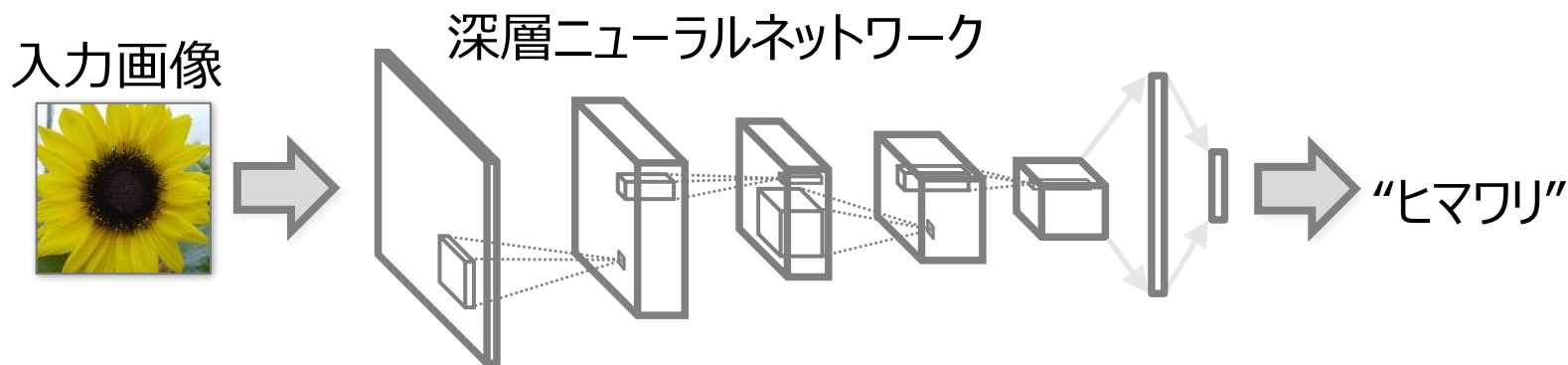
<https://www.desmos.com/calculator/lozdpofklf?lang=ja>



最近のAIの正体： 深層ニューラルネットワーク

深層ニューラルネットワークとは？

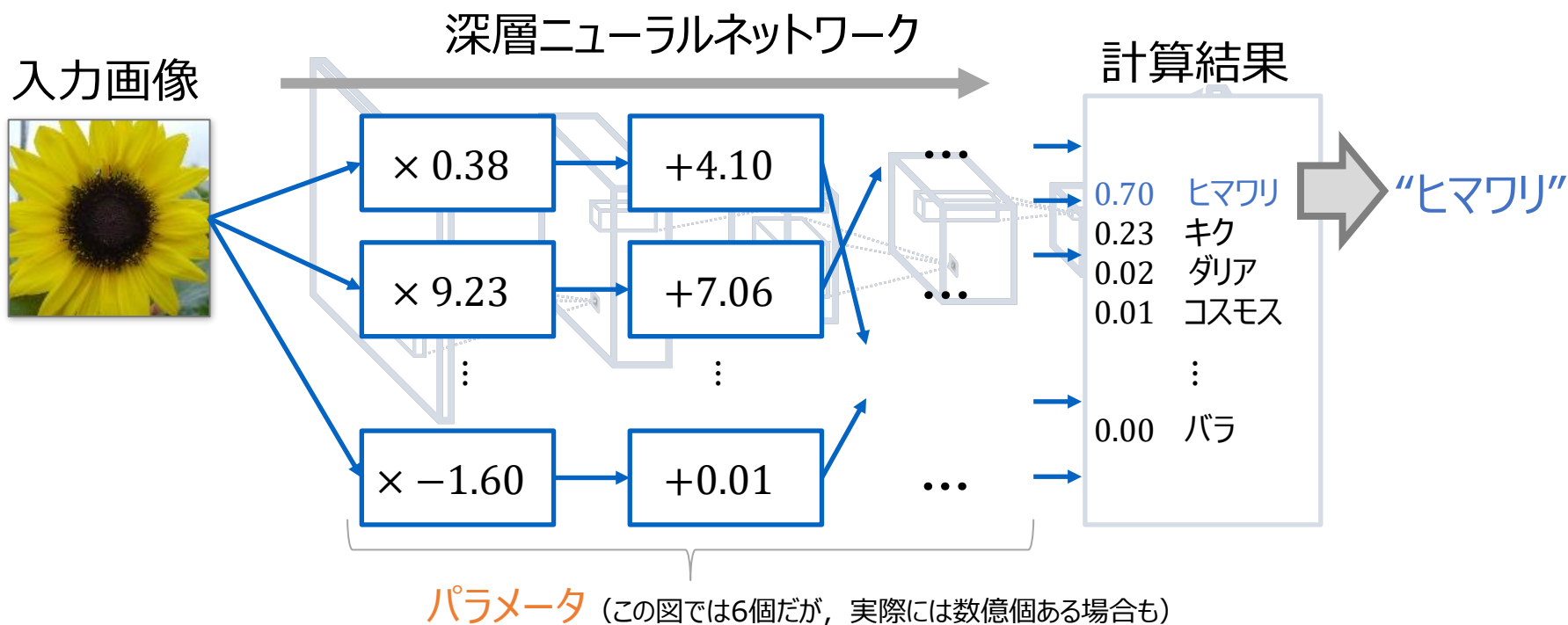
- 最近主流のAI
 - 考え方自体は1970年代から存在
- 予測・認識・生成など，様々な課題に利用できる
 - 画像認識の場合



- 大量データと学習(「深層学習」と呼ばれる)によって，非常に高性能なAIを実現

深層ニューラルネットワーク, どのように動くかを非常にざっくり説明すると...

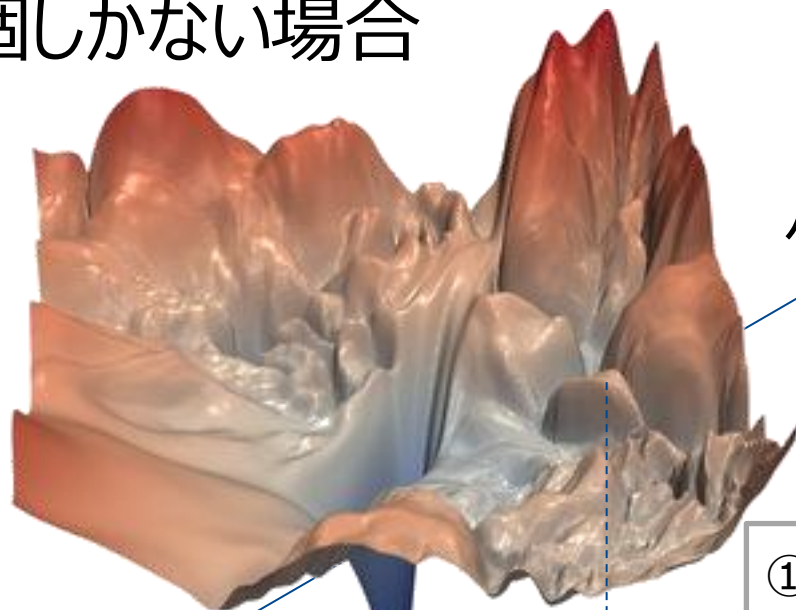
- 入力データに対して, 様々な数を「かけたり」「足したり」することを何回も繰り返すことで, 最終的な答えが出てくる
 - この「様々な数」= パラメータ (よくわからない人は次のスライド)



これらのパラメータの値を適切に調整すること、それが深層ニューラルネットワークの学習(深層学習)

● パラメータが2個しかない場合

ニューラルネットワークの**ダメ具合**
(期待する答えとの誤差)



パラメータ2



③ベストな性能を
与えるパラメータに
調性できた

①最初の状態.
性能イマイチ

②ダメ具合が減るように(=性能が
よくなるように)ちよつとずつ調整

パラメータ1

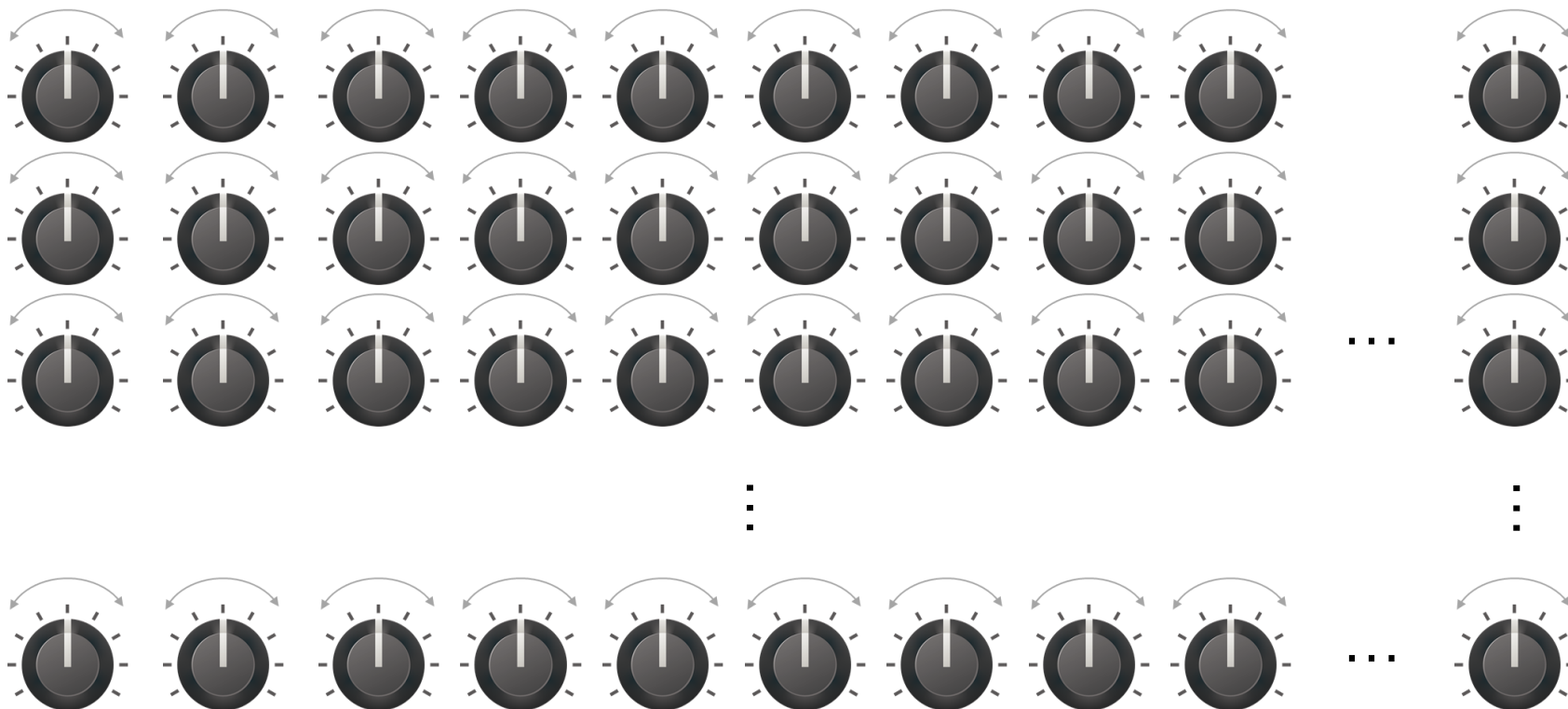


<https://github.com/tomgoldstein/loss-landscape>

[Li+, "Visualizing the Loss Landscape of Neural Nets," NIPS2018]



実際にはパラメータは2個では済まない
数万～数億個ある場合も

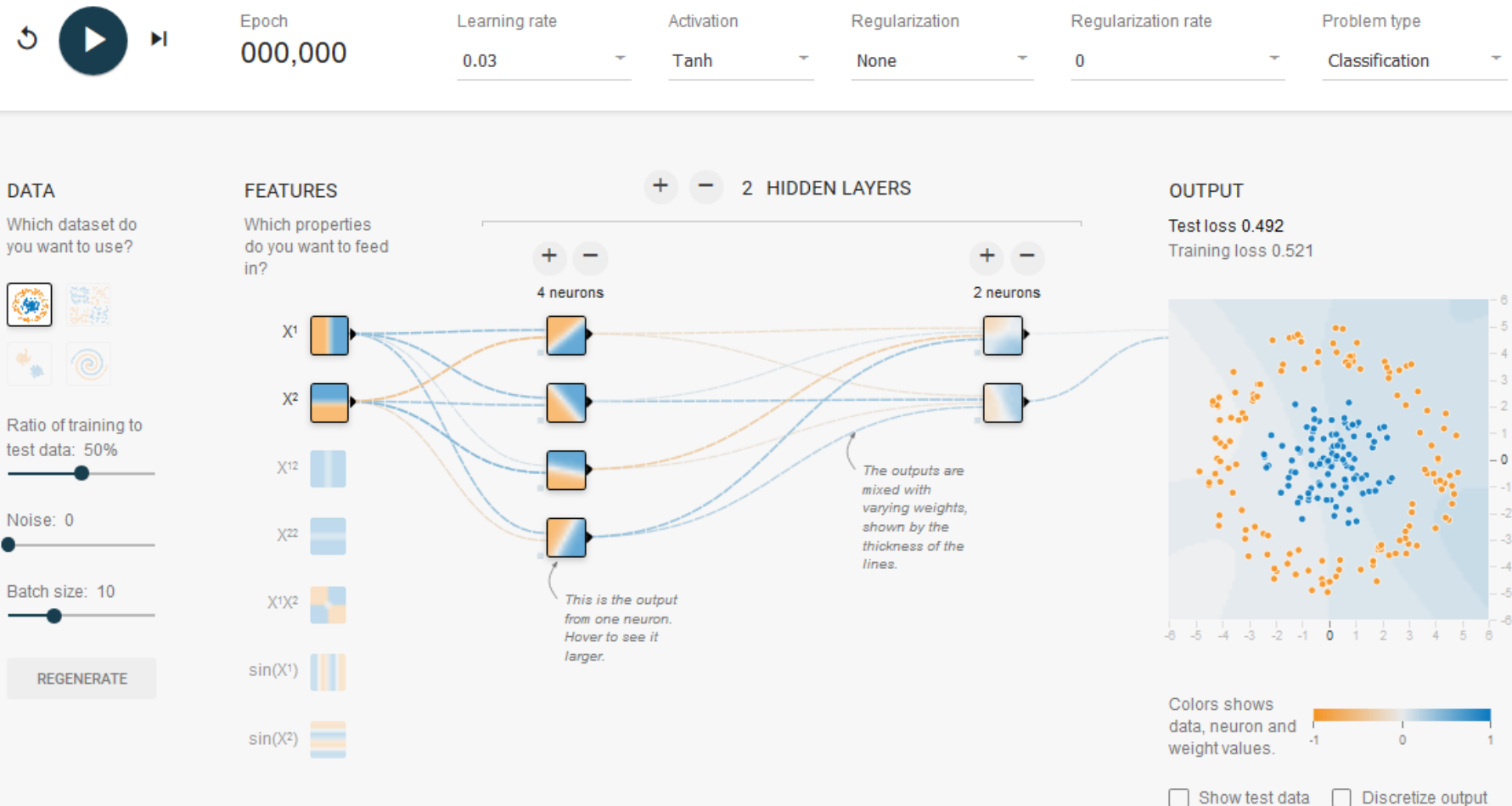


人間には無理！ コンピュータにお願いして、
ベストな「調整」を行ってもらうしかない！

余談：深層ニューラルネットワークの挙動を理解したい人におすすめ

google playground

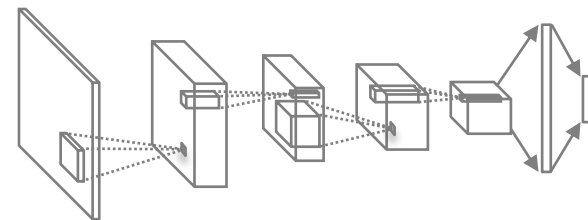
<https://playground.tensorflow.org/>



AIの性能が、最近劇的に向上した3つの理由

- 深層ニューラルネットワークの利用

- パラメータ数が膨大（数千万～数億個のパラメータを持つ）
＝調整の自由度が非常に高い
→うまく調整できれば、相当複雑な問題でも扱える！



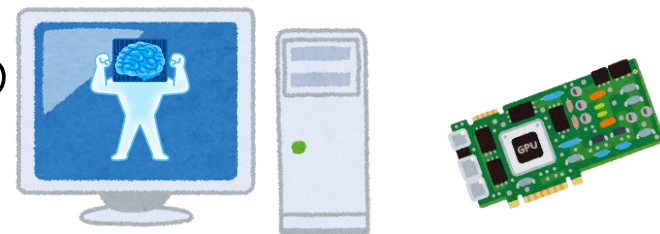
- 非常に大規模なデータが手に入るようになった

- 深層ニューラルネットワークの大量のパラメータを調整（＝学習）するのに十分なデータが手に入る



- コンピュータのパワーアップ

- 大量のデータを用いた深層ニューラルネットワークの学習が可能になった



前ページの「3つの理由」をさらに加速させた 「オープン化の波」



- オープンソース

ソース = source-code = プログラム, のこと

- 頑張った作った最先端のプログラムを, 誰でも使えるように公開してくれているもの. 無償のものも多い

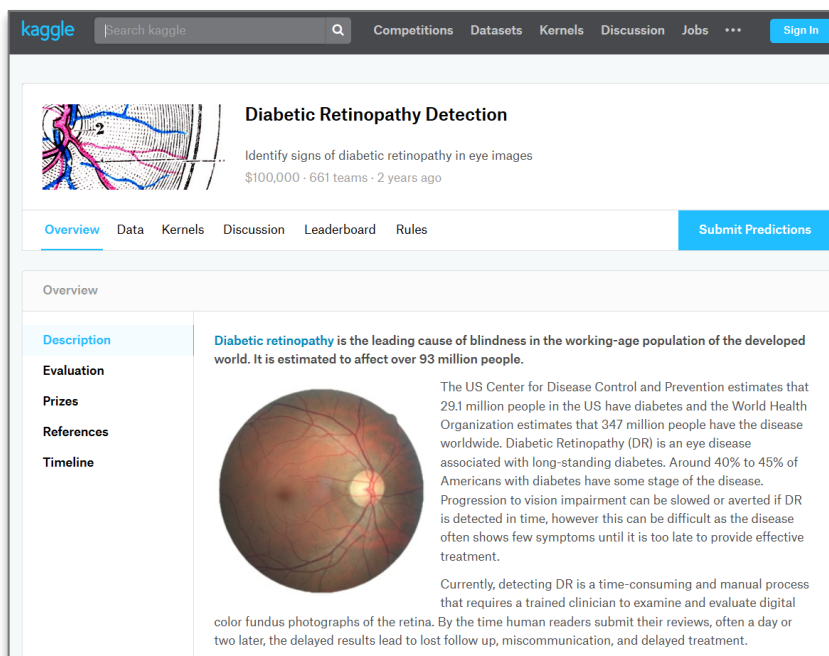
- オープンデータ

- 頑張った集めたデータを, 誰でも使えるように公開しているもの
- 例えば, 画像データや, 自治体などのデータ
- それを使って, 機械学習なども気軽にできる

- 無料で使えるクラウドコンピュータ

- webブラウザからログインすれば自分のコンピュータのように使える
- Google Colaboratory (2025年5月現在無料)

オープン化によるwin-win (みんなハッピー) な例： 糖尿病性網膜症の画像診断AIコンペ (2015)



1. 医師が画像データを集めて公開
2. 世界中のAI専門家がそのデータを使って診断精度を競った
3. 高精度な診断AIを実現 (上位者には賞金も)
4. 医師側はそのAIを使うことで、診断自動化の恩恵を受けた

重症度診断付き
(5段階)画像が
35,000枚!

661チームが参加



賞金も!

1位: \$50,000
2位: \$30,000
3位: \$20,000



サンプルデータの例

今のAIにできないこと

「人間にとって代わる」なんて、まだまだ先か！？

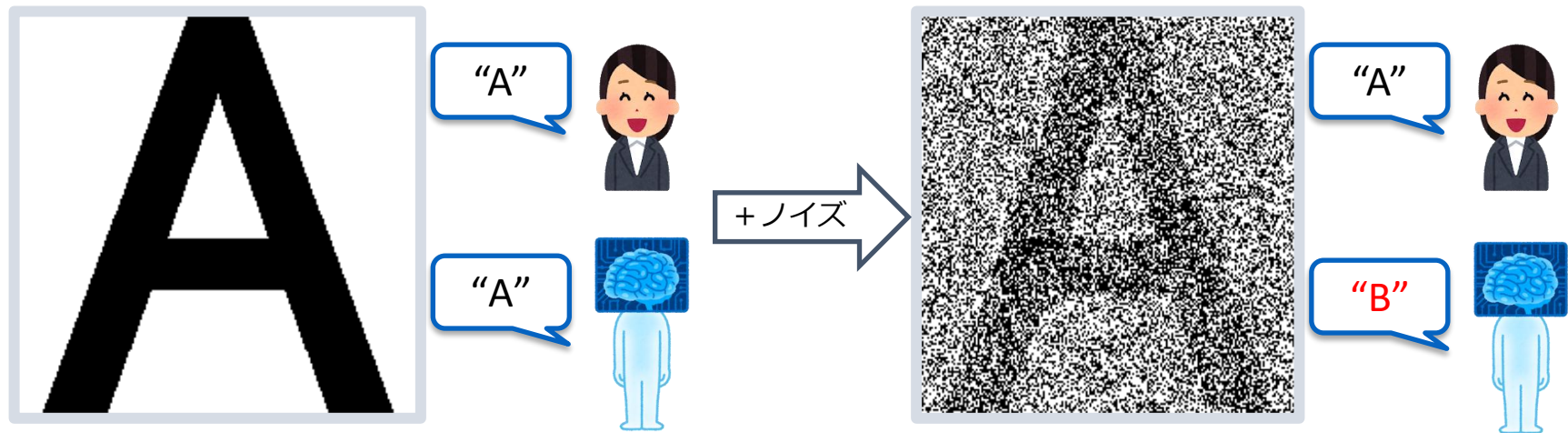
現在のAIは全く万能ではない！

- 実現できているのは「特化型AI」のみ
 - 対話生成や画像認識など，特定のことしかできない
 - ただし，その「特定のできること」の性能は非常に向上していて，下手すると人間以上
 - 汎用AIはまだ研究途上
- 十分なデータがなければ正しく動かない
 - プライバシーにかかわるデータは集めづらい
 - 大地震などの希少データは収集不可能
- 他にも次に挙げる問題が存在（詳細後述）
 - ケース① 敵対的事例
 - ケース② フレーム問題
 - ケース③ 判断根拠が不明確

特化型AIですら全く万能ではない

ケース① 敵対的事例の存在

- 人間だと絶対起こさないような誤りを起こす
- AIが間違えるような意地悪な「ノイズ」を加えると...

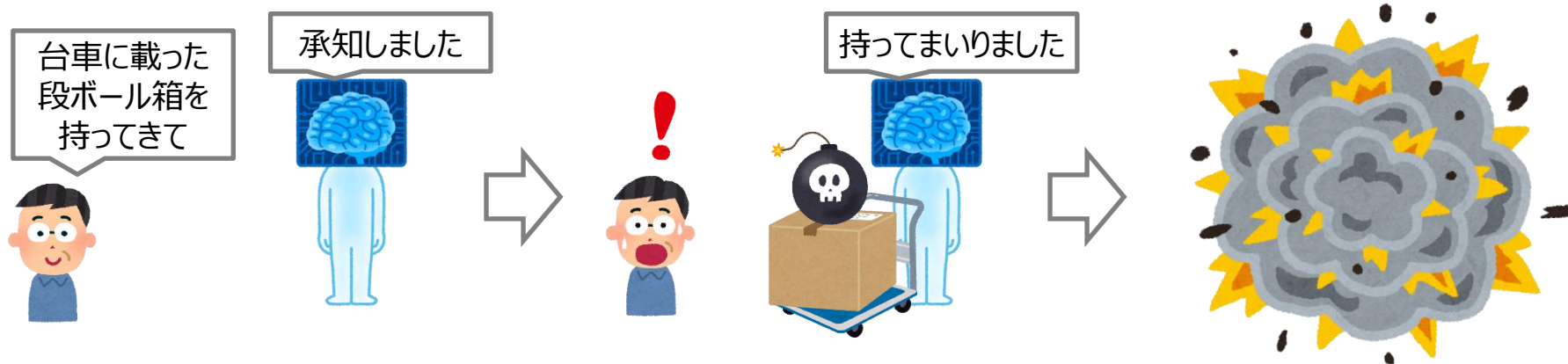


- 「習っていないものについては融通が利かない」ことを意味
 - 上の例で人間は「ノイズは不要」という「常識」を使って正解する

特化型AIですら全く万能ではない ケース② フレーム問題



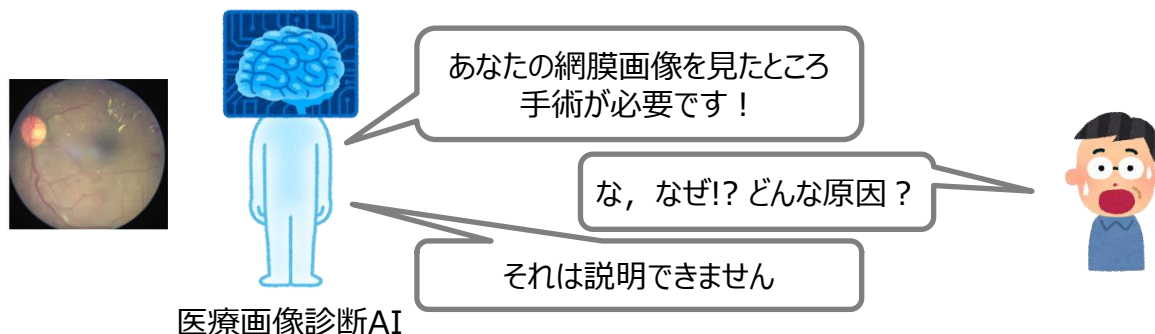
- 特定の範囲のことしか「考えが及ばない」
 - 有名な例：



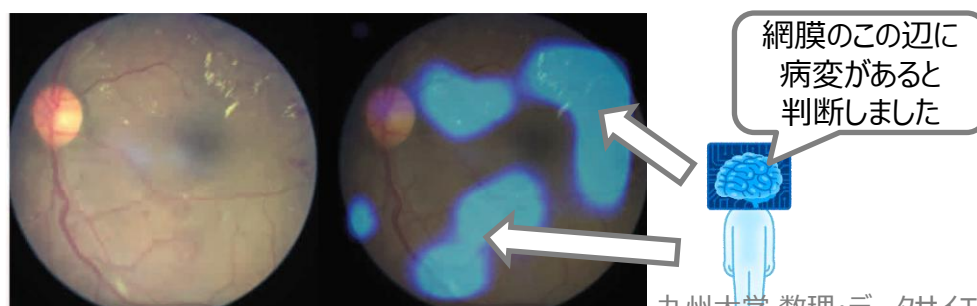
- 色々な可能性を考えたらいいじゃん？ →無理，キリがない！
 - 爆弾が置かれていたら？ 水の入ったコップが置かれていたら？
猫が寝ていたら？，注文用紙が置かれていたら？，箱が壊れかけてたら？...
- 人間がフレーム問題をどう回避しているのかは依然不明

特化型AIですら全く万能ではない ケース③判断根拠が不明確

- AI(深層ニューラルネット)の判断根拠を見出すのが難しい



- 現在この問題を解決するべく explainable AI（説明可能AI）の研究が数多く行われている
 - 例えば「画像中のどの領域に注目して判断したか」を推測する



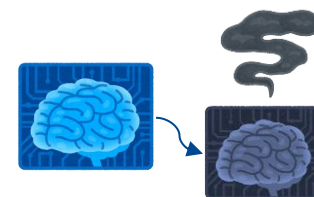
参考：昨今の生成AIは高精度すぎて 様々な社会問題を引き起こしている

いずれの問題についても、法的対応や技術的対応について世界中で議論中（2025年現在）。
今後も状況は変わりうることに注意。

- 著作権
- 誤情報・偽情報
- 有害コンテンツ
- 差別
- データセキュリティ
- AIカニバリズム

生成AIのクオリティが高くなれば高くなるほど深刻になる問題

生成AIの学習に用いるデータに関する問題

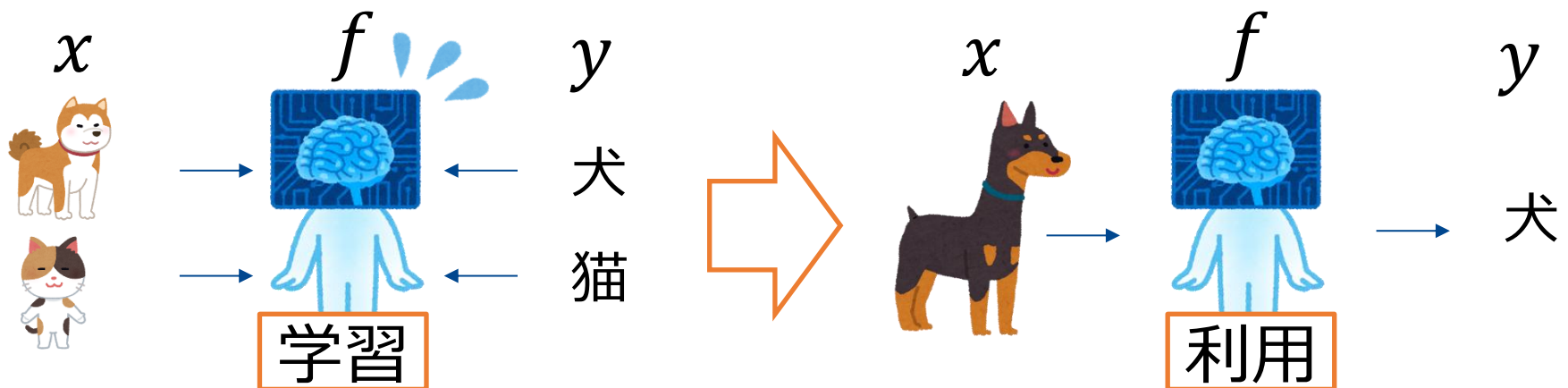


付録

教師あり機械学習

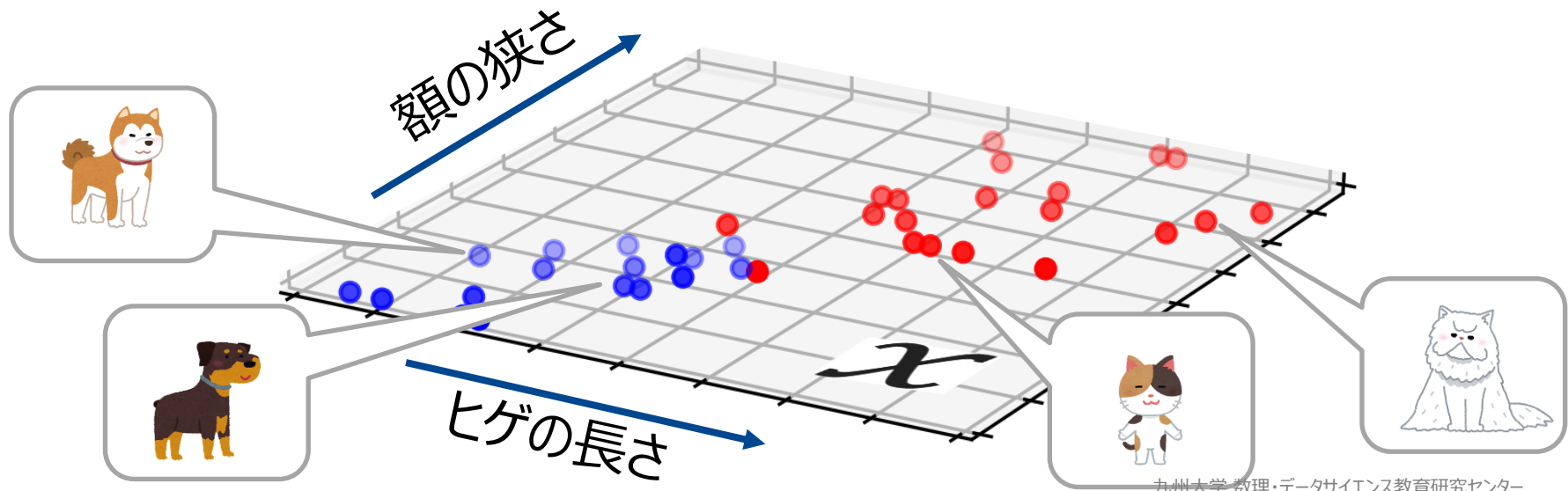
教師あり機械学習とは

- 「この x に対しては、こんな y を返してほしい」という「例」に基づいて、関数 $y = f(x)$ を自動的に定める方法
 - 「例」(学習サンプル)は多数必要
- 「画像認識」ならば...



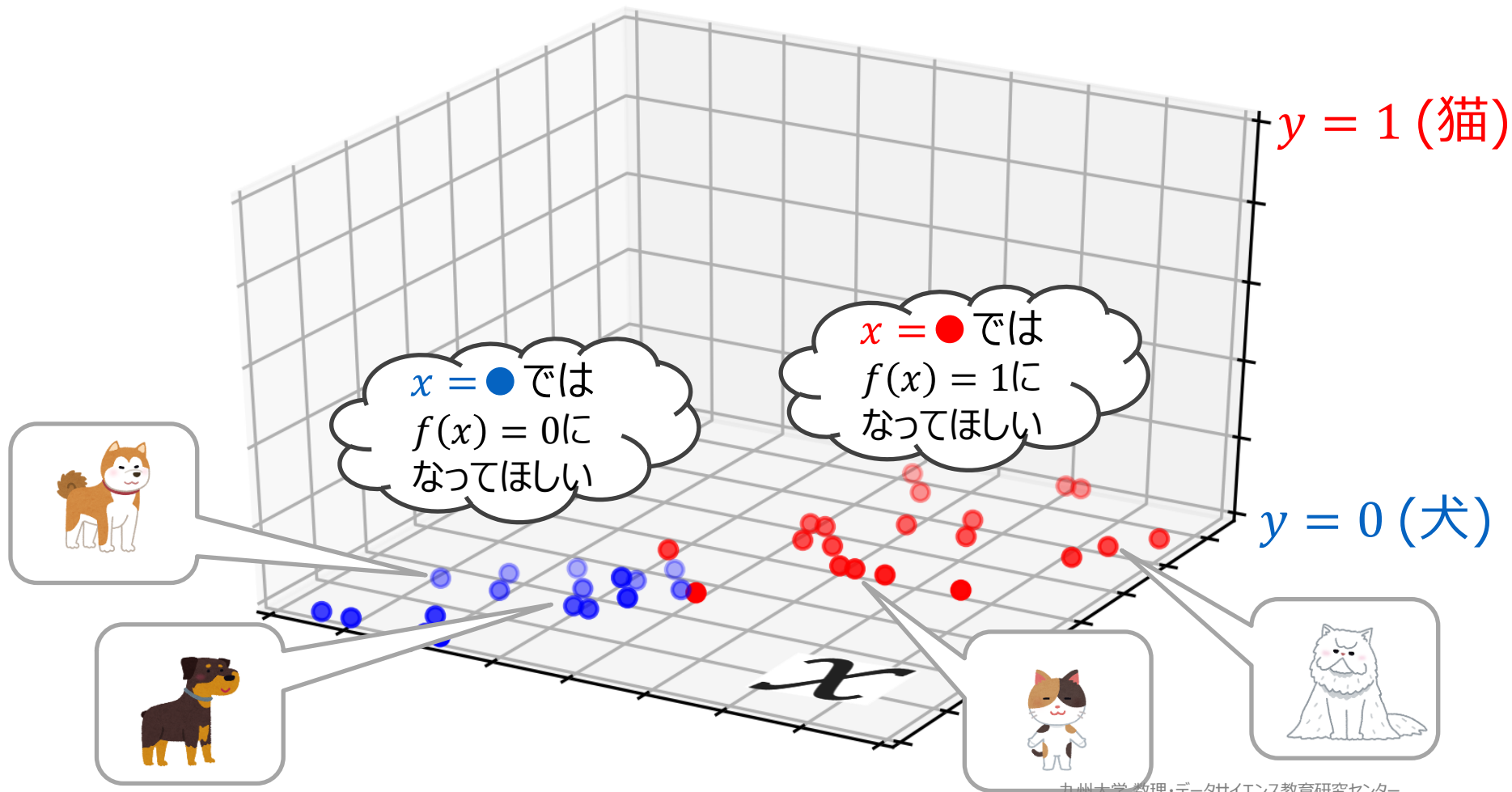
教師あり学習～図で表現

- こういう風に犬と猫のデータが与えられたとする



教師あり学習～図で表現

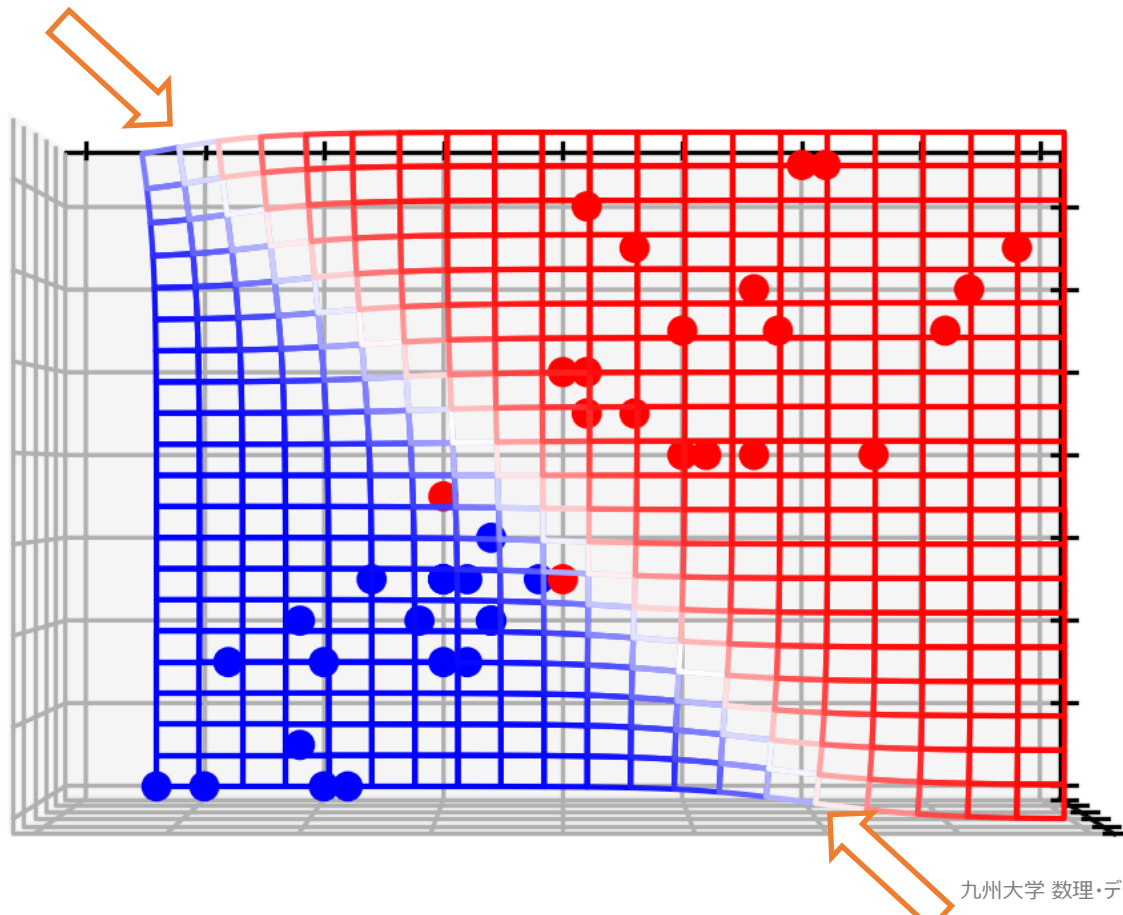
- $f(\text{犬}) = 0, f(\text{猫}) = 1$ となるような関数 f を見つけない



-
- A 3D plot illustrating a non-linear decision boundary. The plot shows a curved surface separating a blue region (labeled $y=0$ (犬)) and a red region (labeled $y=1$ (猫)). The surface is labeled $y=f(x)$. Data points are scattered on the xy -plane, with blue dots in the blue region and red dots in the red region. Callouts show a cat icon for the red region and a dog icon for the blue region.

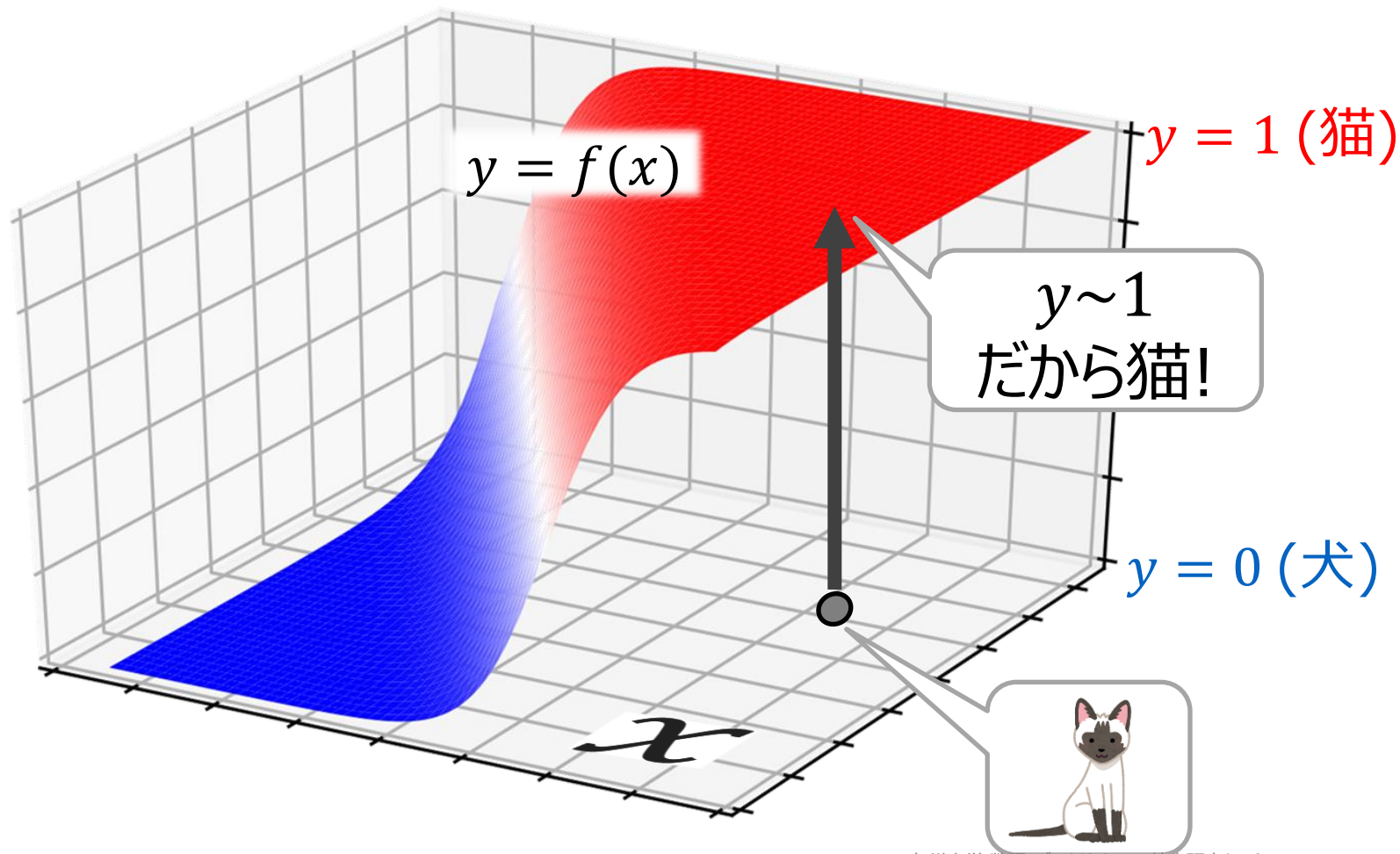
教師あり学習～図で表現

- 上から見ると、犬と猫の境界線がわかる
 - この図の場合、境界線はまっすぐ(直線)



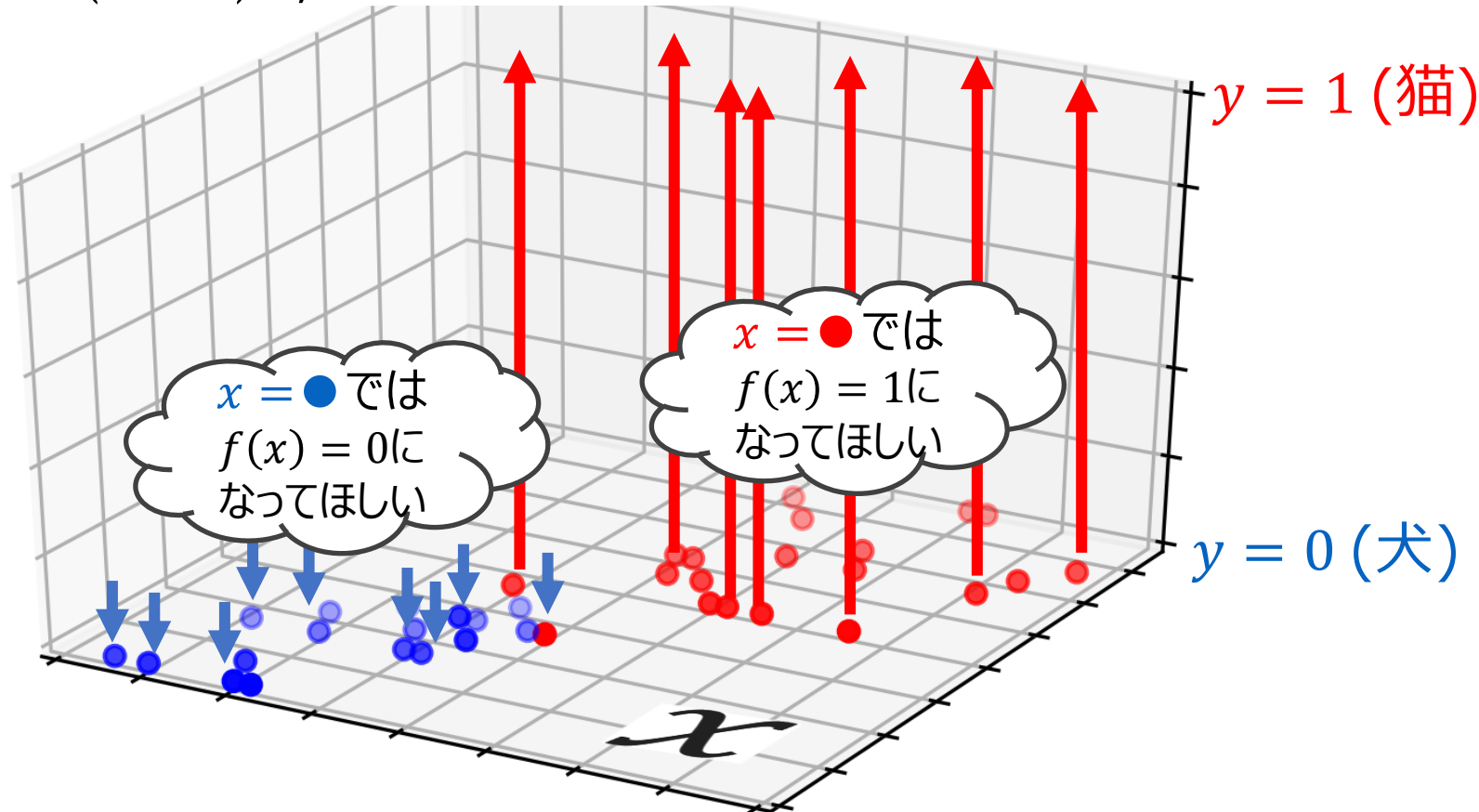
教師あり学習～図で表現

- この f さえあれば「初めて見る x 」も犬か猫か認識できる



何が「教師」だったのか？

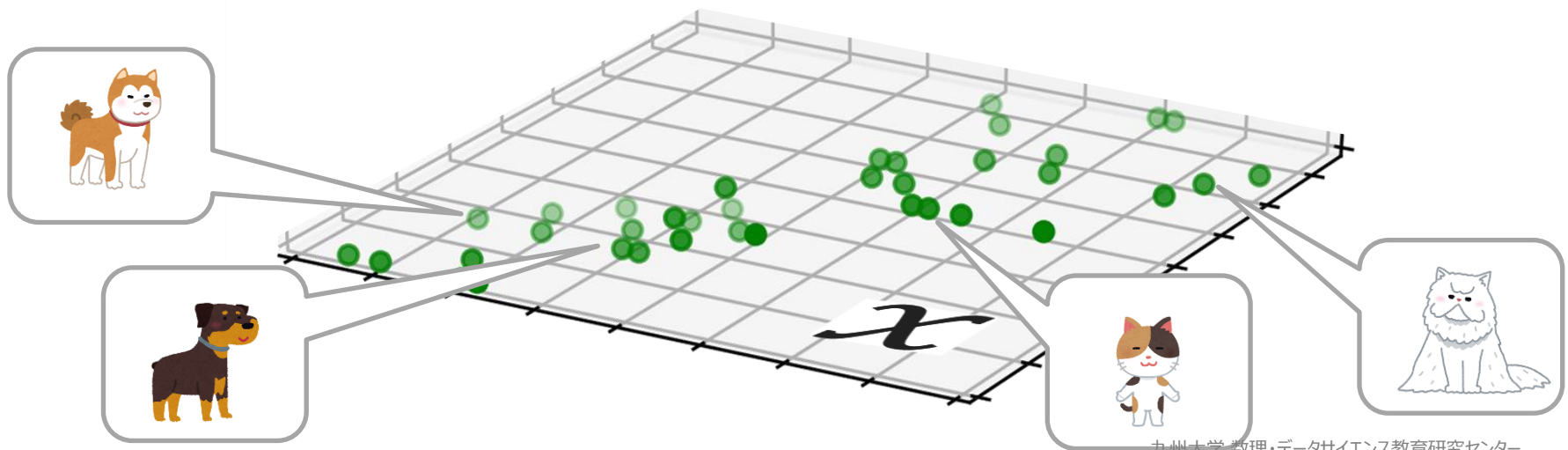
- 各データは「犬」か「猫」かわかっている
- すなわち人間(=教師)が、各データの正解を教えている



教師なし機械学習

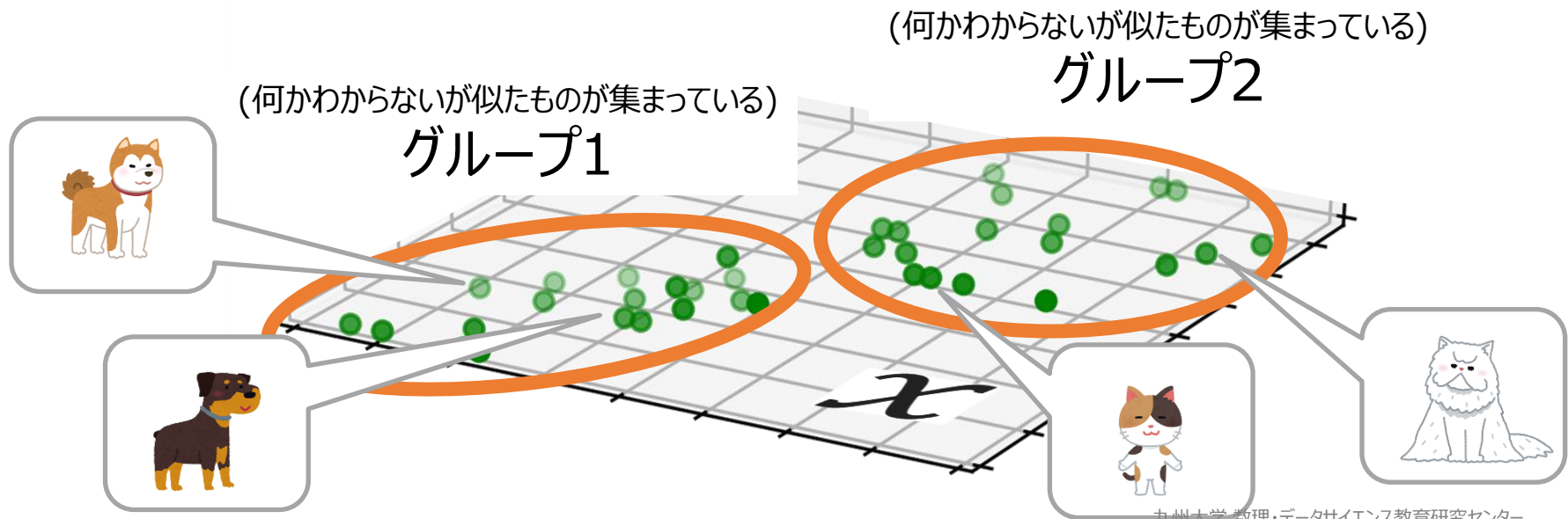
では教師が無い状態というのは？

- 計算機は、どのデータが犬か猫か、知らされていない
- なので、 $y = f(x)$ を作ることはできない
- なので、画像認識のための機械学習はできない
- では、何もできないのか？



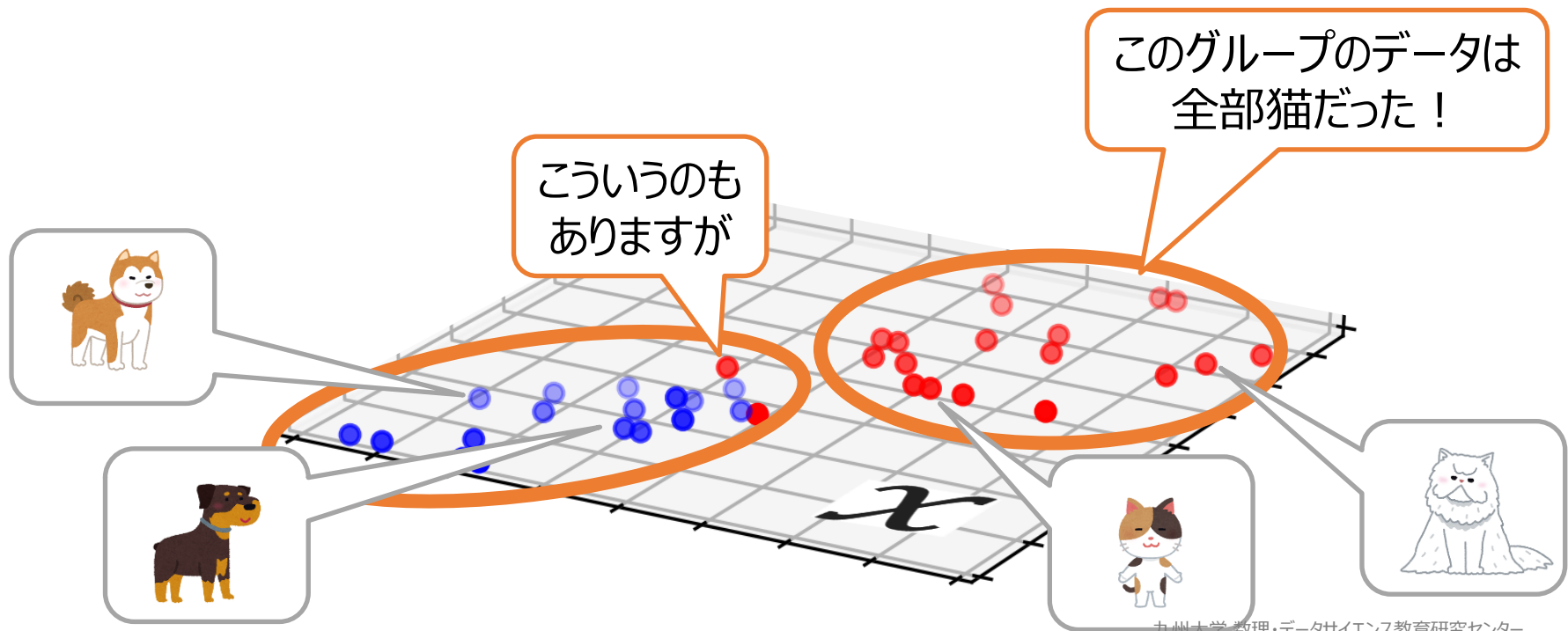
「教師無し学習」ならできる

- 近くにいるものどおしを「グルーピング」すること
 - すなわち（前にやった）「クラスタリング」とは，教師なし学習の一種



「教師無し学習」に期待すること

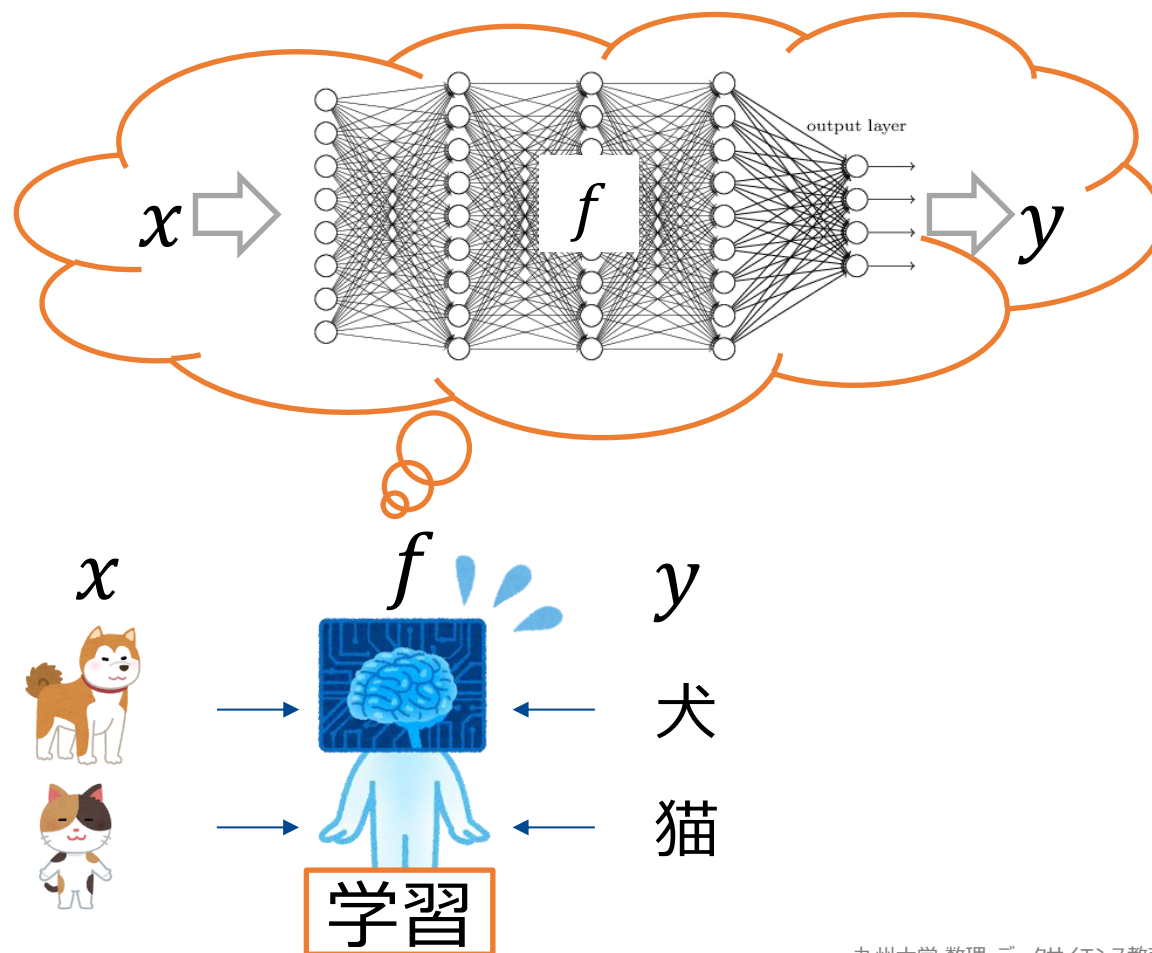
- 「グループの中身を見てみると、ほとんどが同じ部類だった」ら嬉しい！



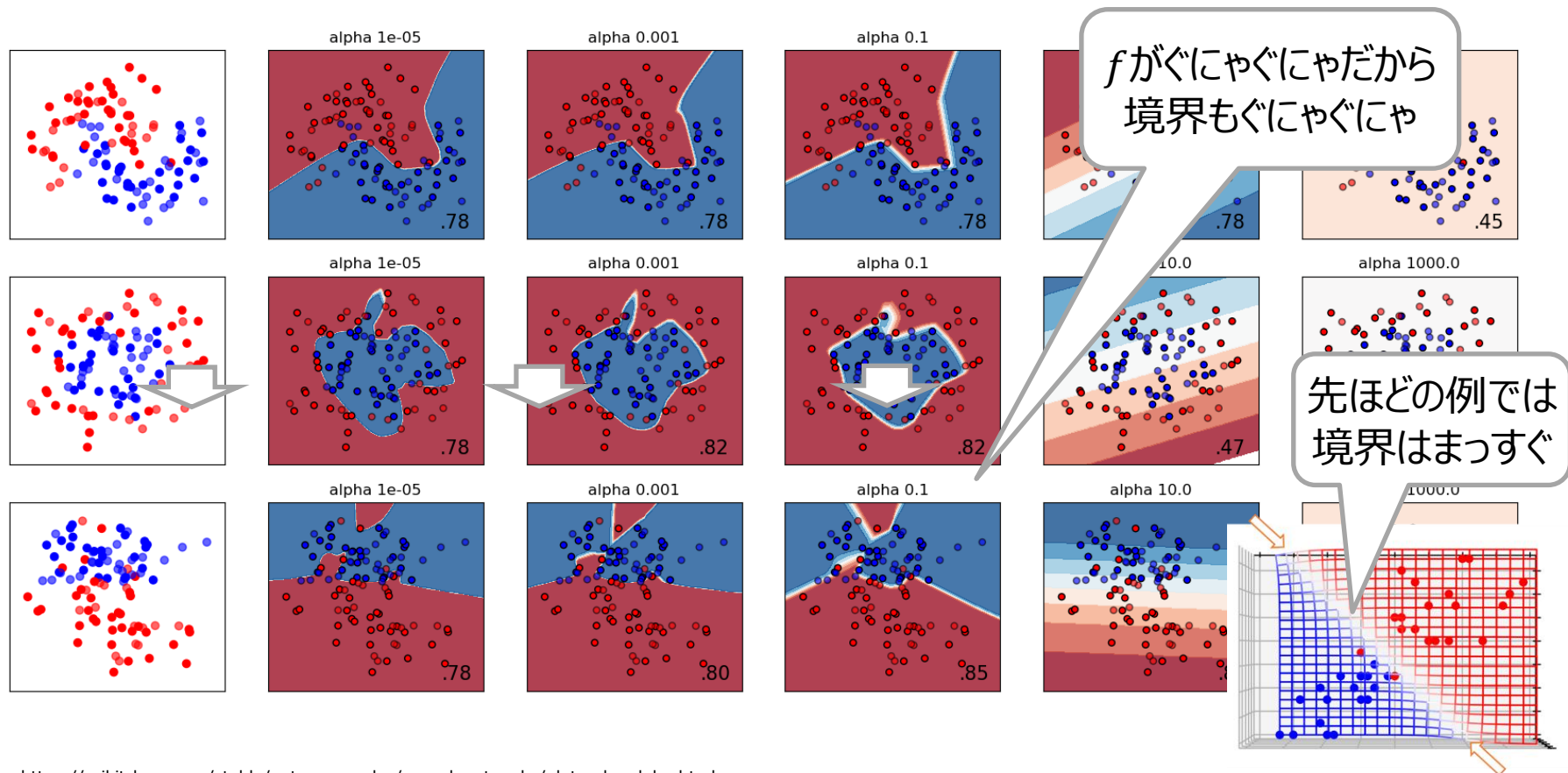
深層学習に関して もう少し

深層学習とは？

- 深層ニューラルネットワークにより関数 f を実現するための機械学習法



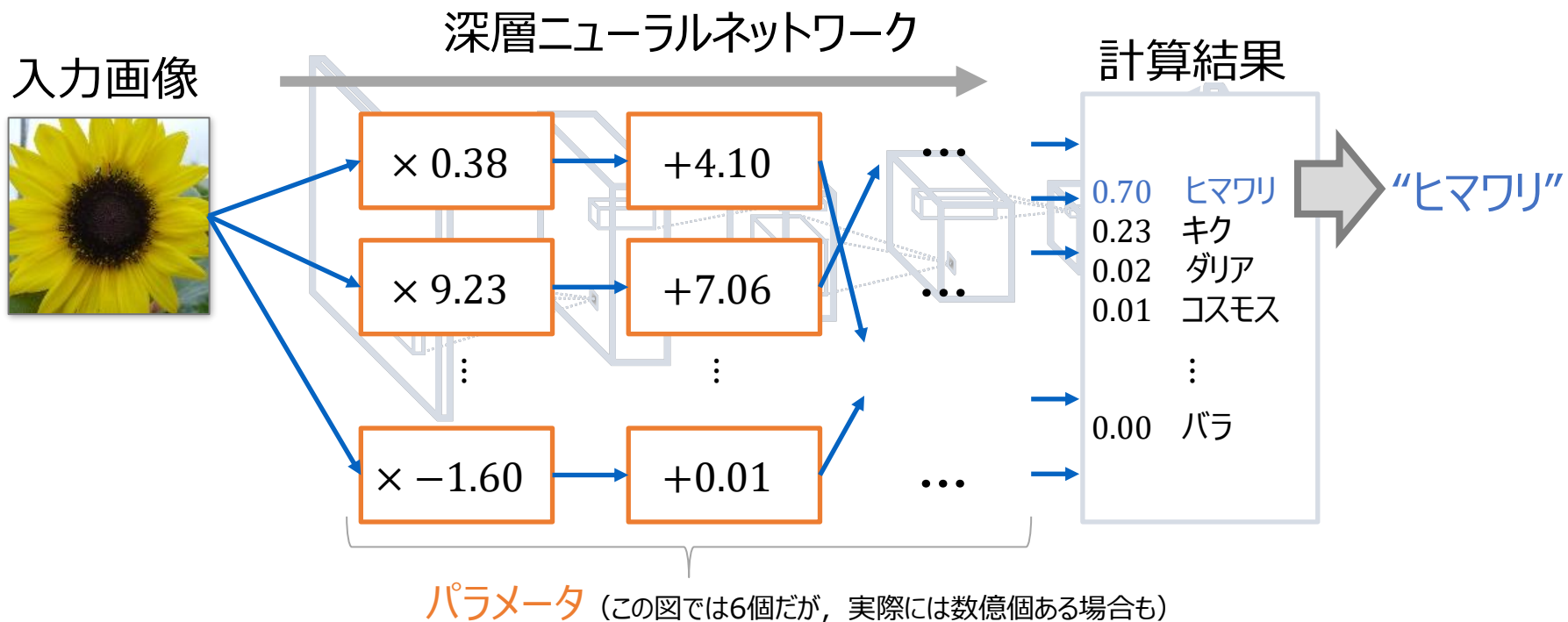
複雑に入り組んだ問題でも分けられる (=難しい認識課題でも、対応できる!)



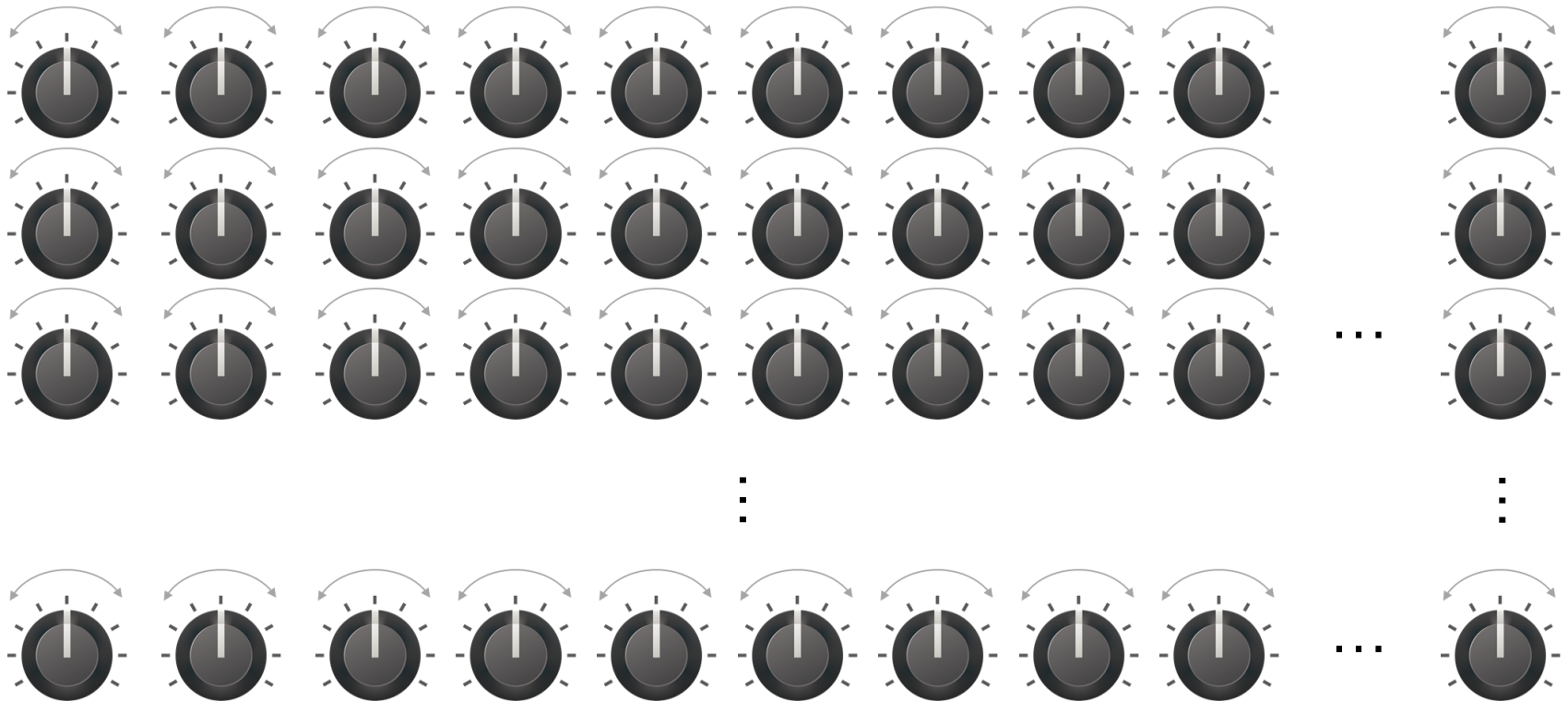
https://scikit-learn.org/stable/auto_examples/neural_networks/plot_mlp_alpha.html

凄そうなことはわかった。
でも、このパラメータを決めてやる必要があるんですよね？

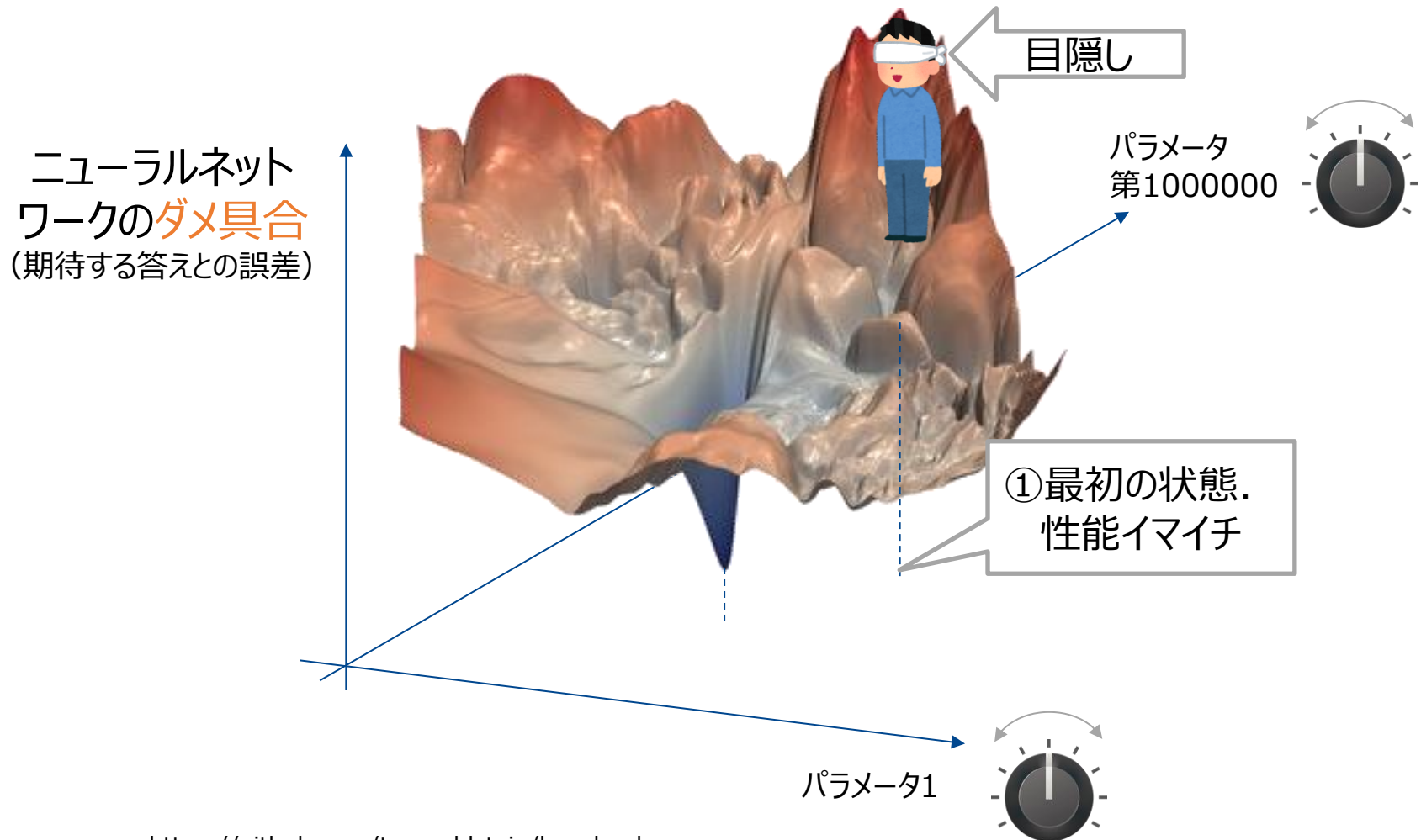
- Yes
- でも、どういう値がベストかわからない。
- さらに言えば、数億ぐらいある場合もある



さて、どうする．数億の「つまみ」(パラメータ)を
全部調整するのはコンピュータだって困りそう



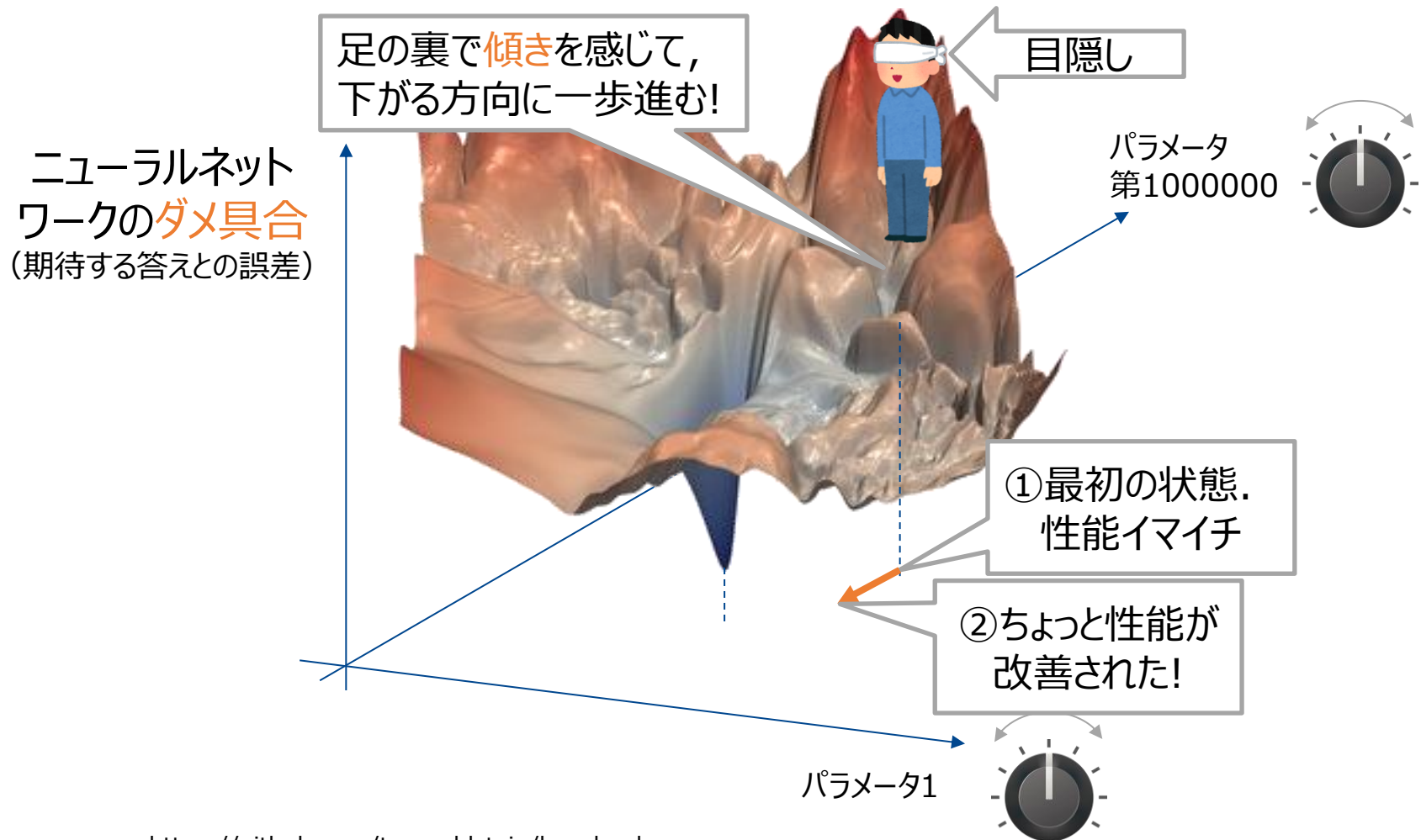
パラメータ数が膨大過ぎて目隠した状況に近い。
このとき、どうやって「ベスト状況」を探す？



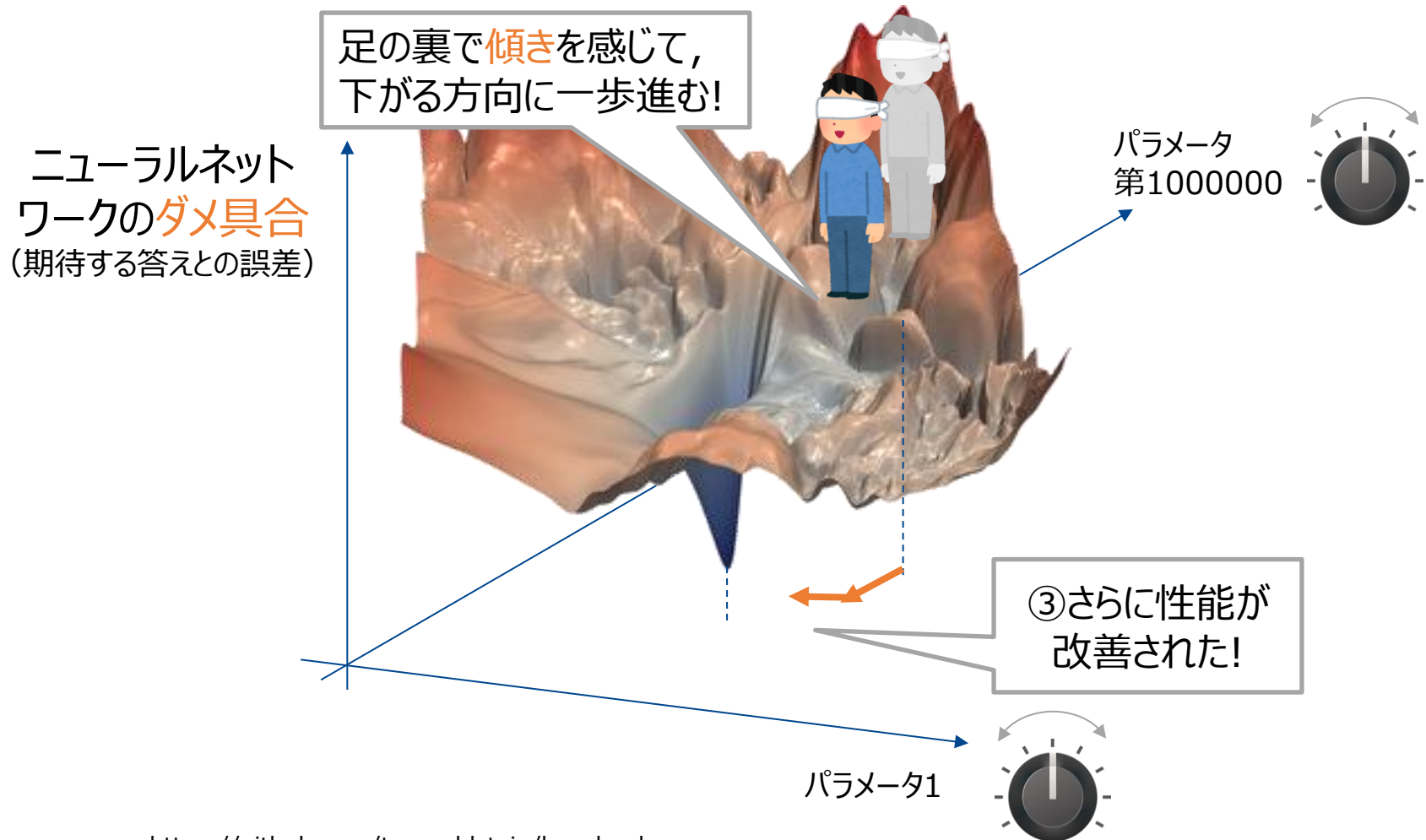
<https://github.com/tomgoldstein/loss-landscape>

[Li+, "Visualizing the Loss Landscape of Neural Nets," NIPS2018]

多分、足の裏で傾きを感じて、下がる方向に進むのでは？

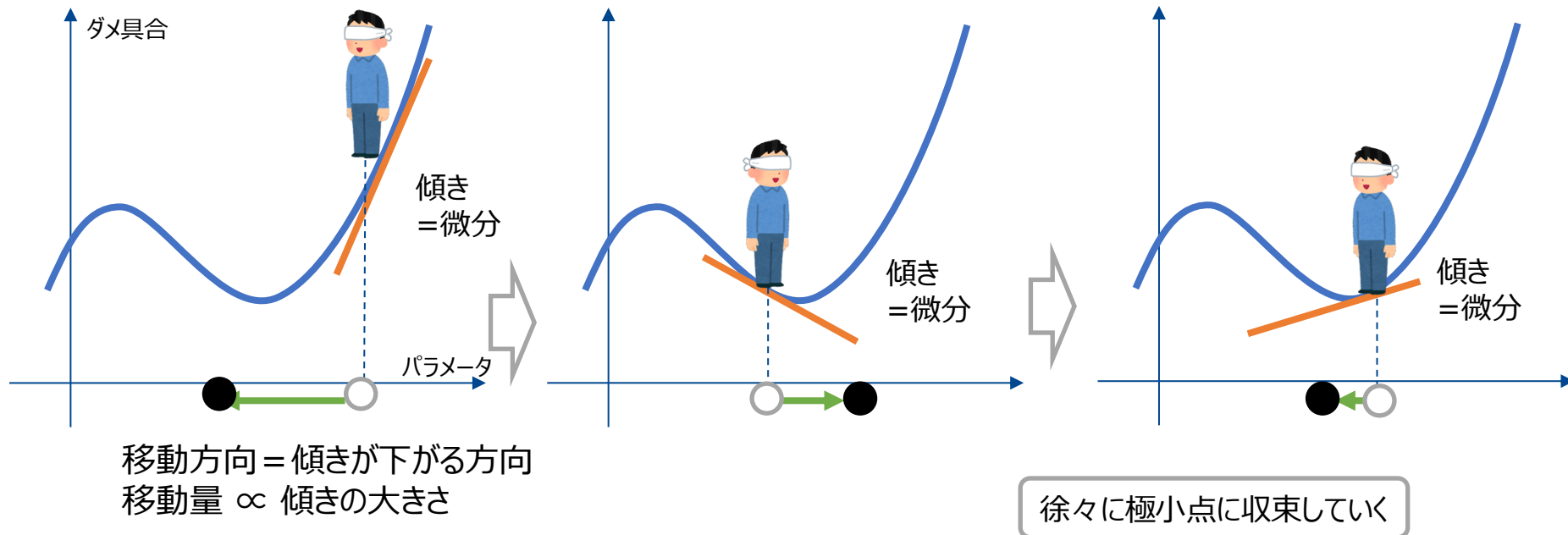


それを繰り返せば、全体像が見えなくても、
少しずつ改善できそう！



この旅の手掛かりは「微分」!!
微分で傾きがわかり、それで動くべき方向がわかる!

- 更に簡略化したケースで見てみる

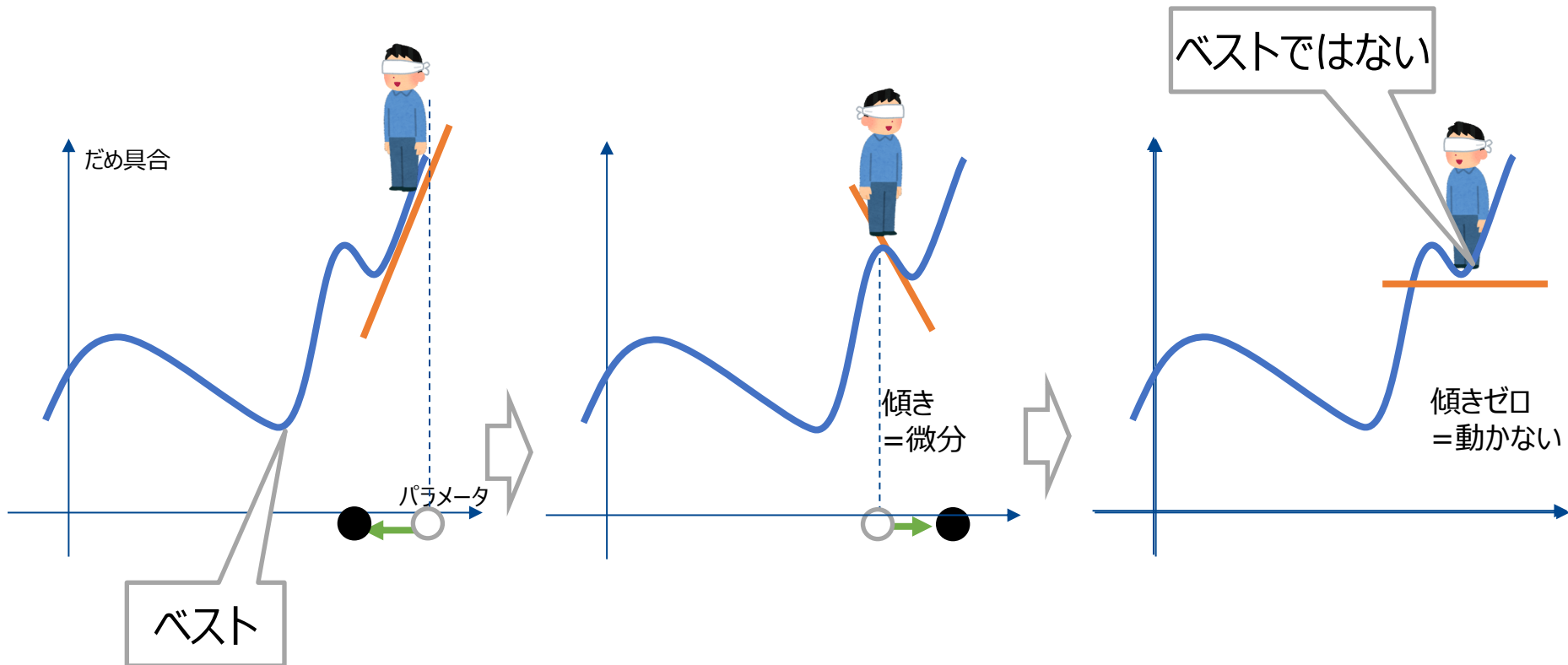


というわけで、微分便利、重要!

(皆さんが微分嫌いでも、皆さんの脳は勝手に微分を使ってる)

気づく人は気づくでしょう。 「現実はそう甘くない」ことを

- あらら、「ベストではないところに引っかかってしまった」



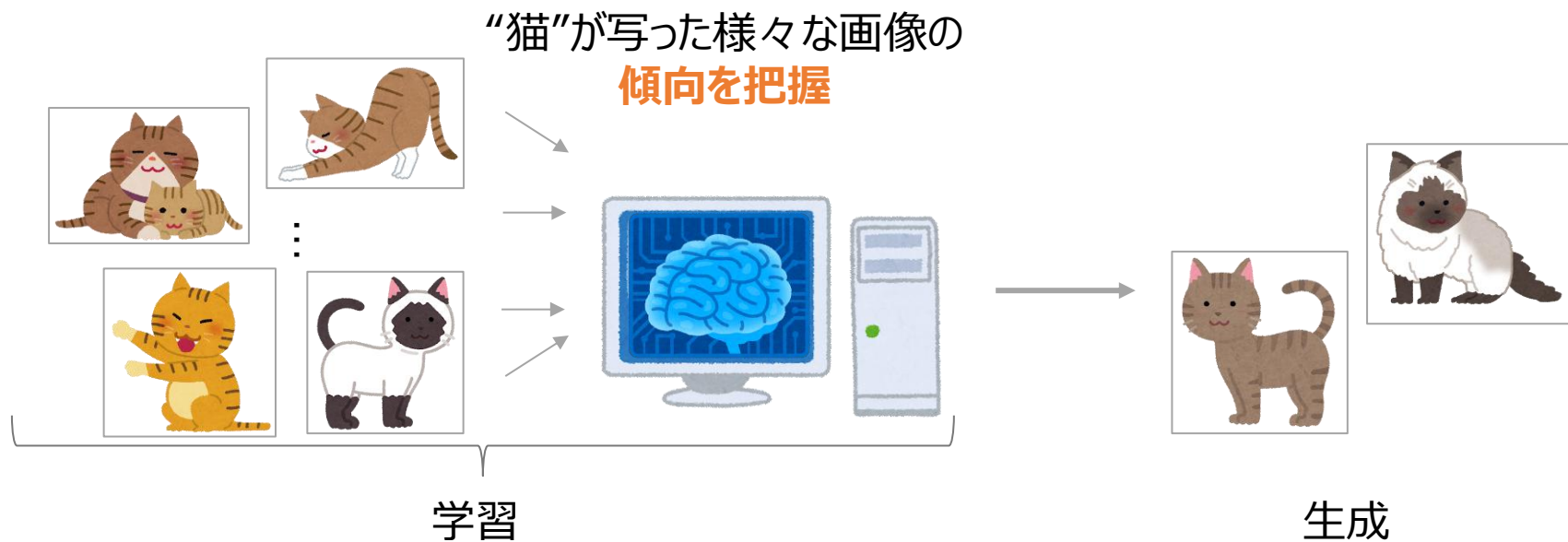
- 深層ニューラルネットワークが、この状況をどう回避しているのかは、世界中で研究されている（完全には解明されていない）

AI・機械学習関連の少し進んだ話

敵対的生成ネットワーク・強化学習・転移学習

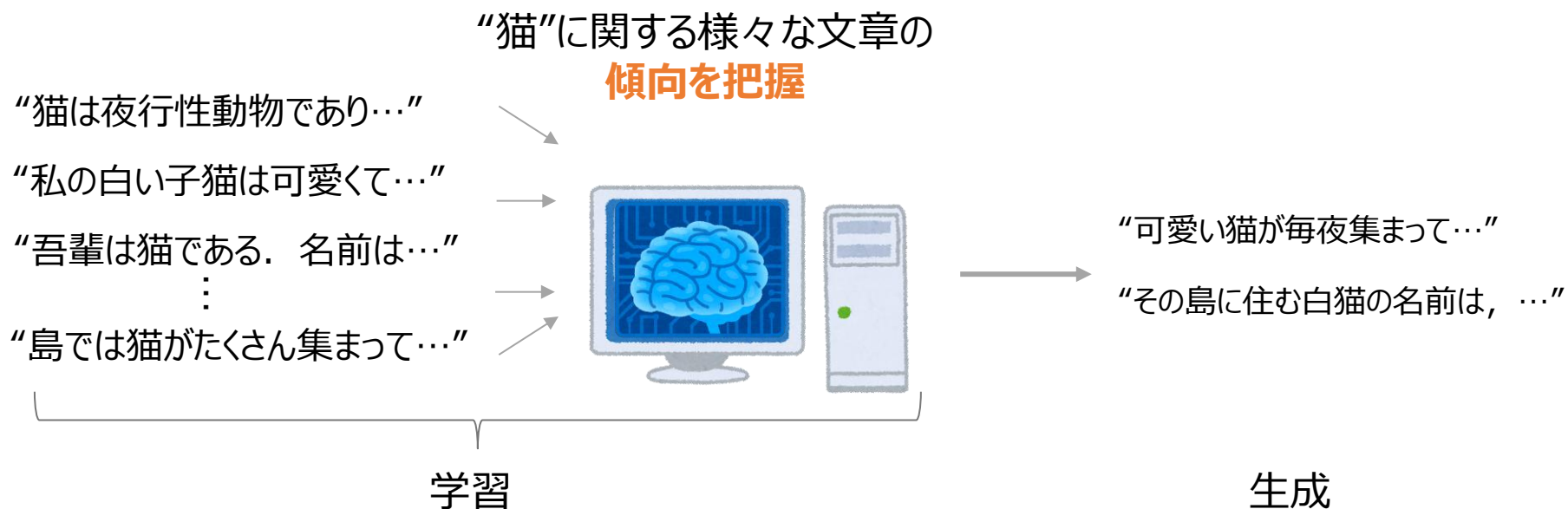
生成AIの原理 (1/3)

画像生成AIの場合



生成AIの原理 (2/3)

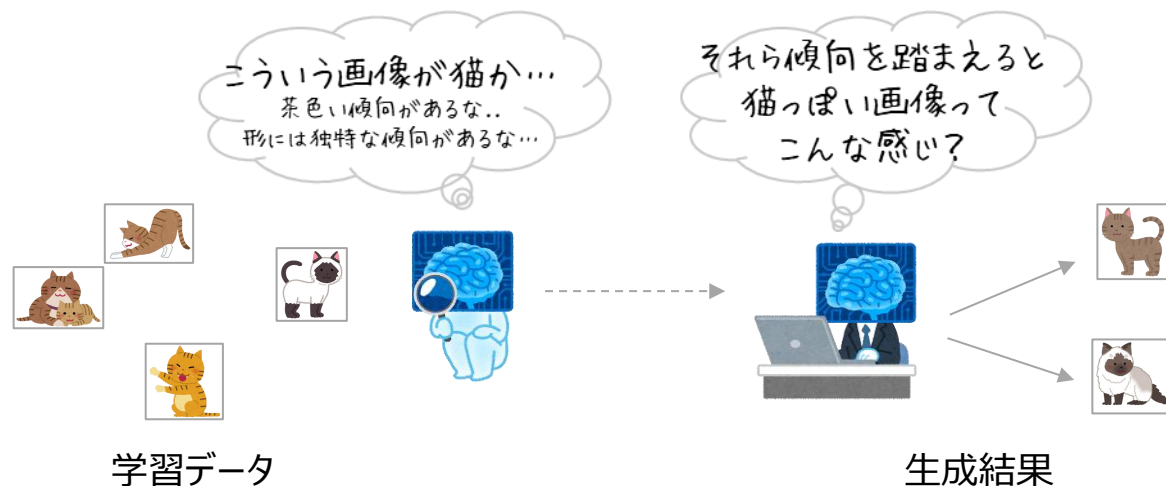
テキスト生成AIの場合



生成AIの原理 (3/3)

まとめると...

- 多くのデータを学習し、「それらとは異なるが、似たデータ」を作り出す
 - 例えば、1万人分の顔画像で学習した生成AIは、「その1万人とは異なるが、実際にいそ
うな人の」顔画像を、無限に作ることができる
- 従って、「学習したデータの背後に潜む傾向」を把握するAI



生成AIは進化している！ 画像生成AIを例に

特定物の生成
例：顔



2014



2015



2016



2017



2018

(Goodfellow 2019)

進化

動画の生成


<https://openai.com/index/sora/>

プロンプトによる任意画像の生成
例：「アボカドの形をした椅子」


<https://www.technologyreview.jp/s/230408/this-avocado-armchair-could-be-the-future-of-ai/>


進化



進化

強化学習：その背後にある考え方



将棋が強くなるには、
「状況を判断する力」が必要



しかし、いきなりその力を
持てるわけではない



試行錯誤で「状況を判断
する力」をつけていく
(=**強化学習**)



運転がうまくなるには、
「状況を判断する力」が必要



しかし、いきなりその力を
持てるわけではない



試行錯誤で「状況を判断
する力」をつけていく
(=**強化学習**)

強化学習：方法

- 現在の場面において、現時点での知識に基づいて、最善と思われる**行動**をする
 - 将棋の例：ある手を指す
- それが長期的に見てよかったか・悪かったか（当然失敗もする）を**評価**
 - 将棋の例：最後まで指し続けて、指した手が失敗だったか成功だったかを評価
- 評価結果に基づいて**知識を修正**
 - 将棋の例：失敗だったらそういう手は指さないようにする
- こうした試行錯誤を**繰り返し**、知識を洗練していく



強化学習：応用

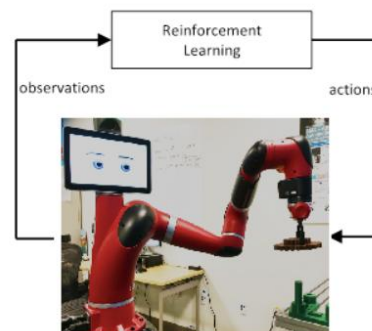
- ゲーム

- ブロック崩しやスーパーマリオ（遊んでみたい人は OpenAI Gym）
- AlphaGo: 2017年に人類最強のトップ棋士に勝ち越す



- ロボット制御

- ロボットの腕(ロボットアーム) でうまく部品をつまむ方法を学習させる



Siemens社

- 自動運転

- アクセル・ブレーキ・ハンドルを制御しながら安全に目的地にたどり着く

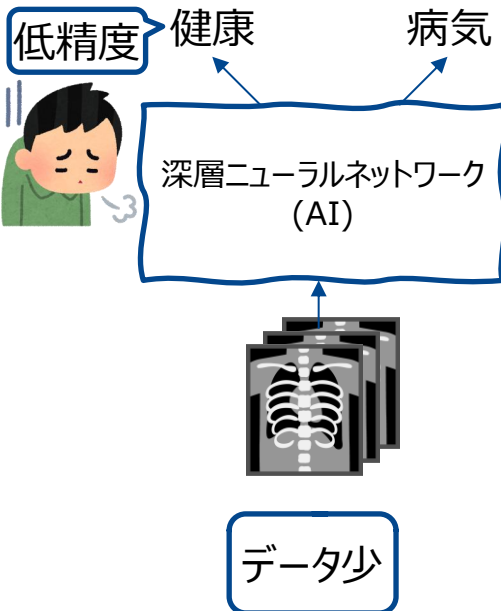


songyanho.github.io

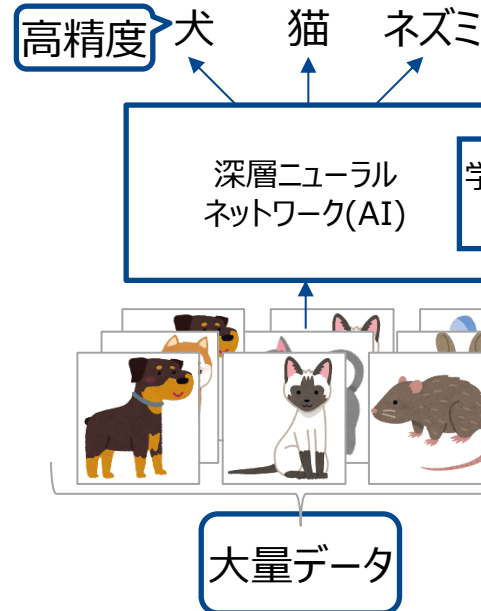
転移学習：考え方と方法

- 少ないデータしかなくても、それでなんとか賢いAIを作りたい
 - そのために、別の大量データで学習した結果（パラメータ）を使いまわす！

少データでは精度が出ない



そこで大量の別データでまず学習



少データで追加学習

