

2. 人工知能の仕組み, 責任ある活用

(コンピューターサイエンス)

金子邦彦

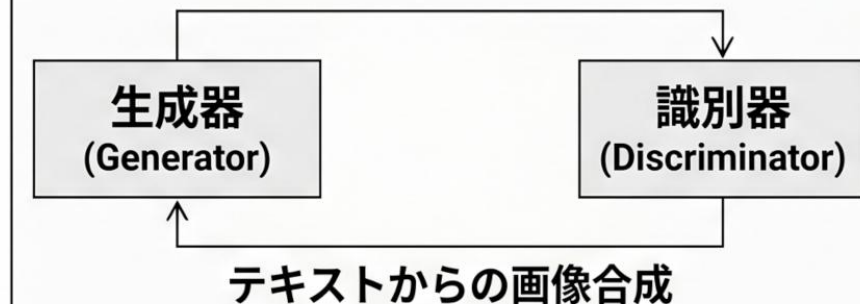


謝辞：この資料では「いらすとや」のイラストを使用している

システムの違い

プログラミング (知的なITシステム)	条件分岐・繰り返し
機械学習	データの学習と予測

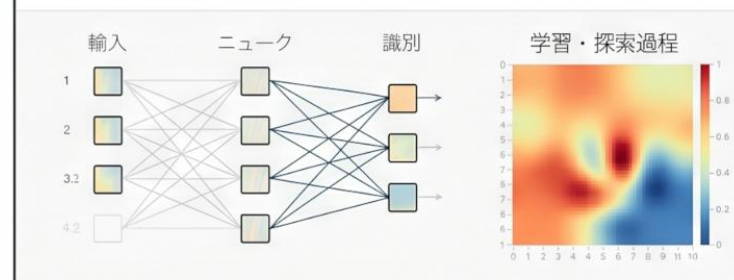
GAN (敵対的生成ネットワーク) の仕組み



責任あるAI活用

- AIの誤りとデータの偏り
- 責任ある判断力

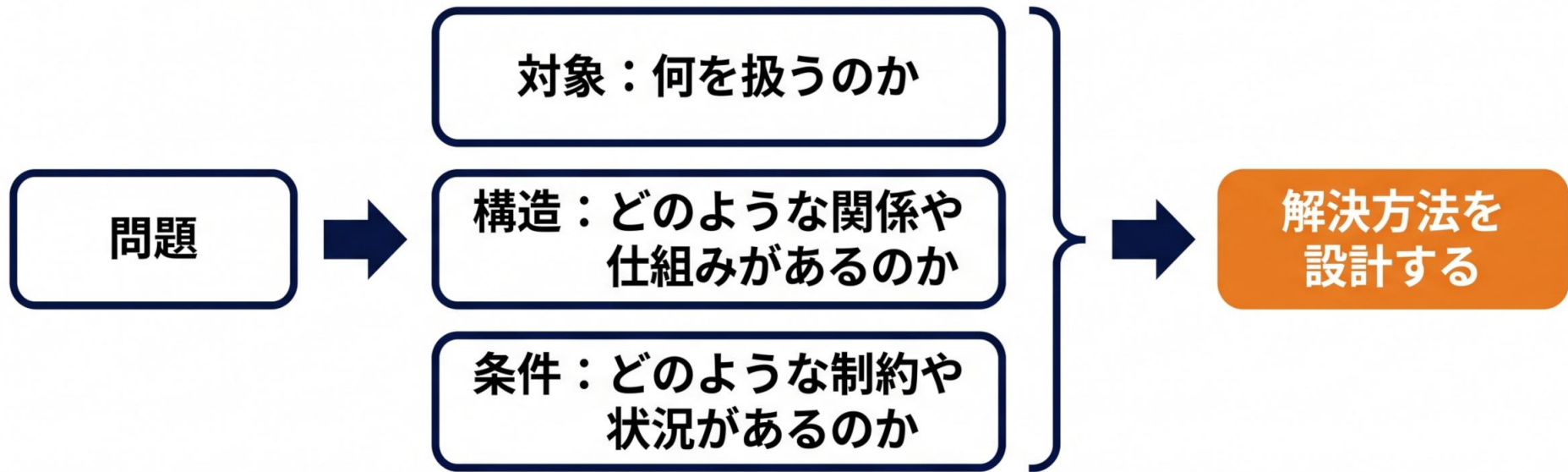
TensorFlow Playground



動作原理の体得

2-1 情報工学と人工知能

情報工学：問題を解決するための「見極める力」



例：キャラクターの自動の動き

対象 → キャラクタ

構造 → 位置と速度の関係、移動のルール

条件 → 画面の範囲、障害物、速度の上限

やみくもにプログラムを書くのではなく、まず対象・構造・条件を整理してから設計する。
これが情報工学で求められる「見極める力」。

情報工学の基本は「対象」、「構造」、「条件」。

人工知能もこの3要素に分けて整理すると理解しやすい

対象

何を扱うのか

- 画像データ
- テキスト
- 数値データセット

構造

どのような関係や仕組みがあるのか

- ニューラルネットワークの層と接続
- グリッド上の通路と壁

条件

どのような制約や状況があるのか

- 正しく分類する
(機械学習)
- 最短経路で到達する
(知的なITシステム)

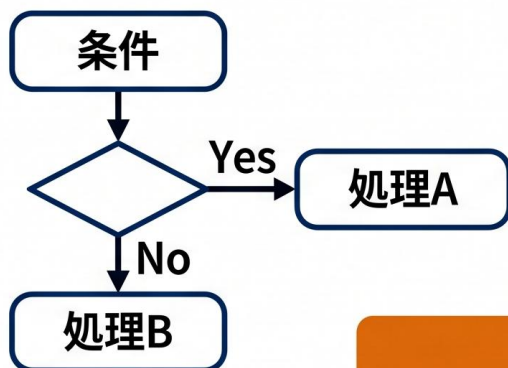
プログラミング基礎とAIの連続性



プログラミングは、AIの仕組みとつながっている

条件分岐

もし～なら～，でなければ～



知的なITシステムの根幹。
専門家の知識をルールとして記述する。
例：もし体温が38度以上なら発熱の可能性／
もし特定のキーワードを含むなら迷惑メール

基礎概念

AI（機械学習・探索）の根幹

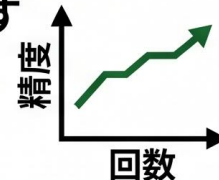
繰り返し

同じ処理を何度も実行する



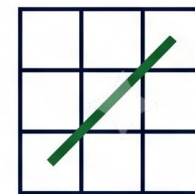
機械学習

訓練データによる学習を何百・何千回繰り返す（1回=Epoch）。繰り返すたびに精度が向上

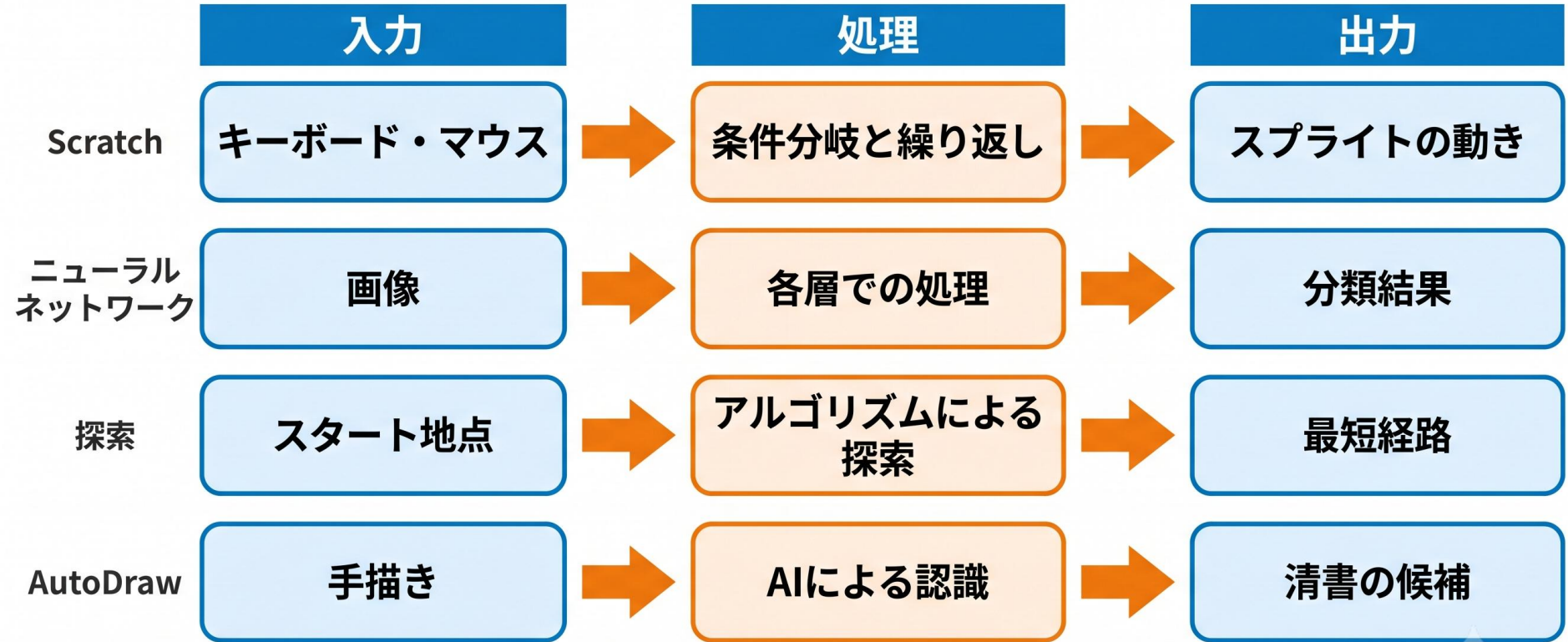


探索

隣接するセルを繰り返し調べ、最短経路を発見



入力⇒処理⇒出力：プログラミングとAIに共通する流れ



「入力 → 処理 → 出力」の流れは、プログラミングでもAIでも共通する情報工学の基本

2-2 AI の基礎

AIの定義

コンピュータが人間のような知的能力を持つことを目指す技術

AIの3要素

① 知能

思考や判断
などの能力

② 知識

情報を収集し、
処理する能力

③ 学習

知的な能力が
向上する能力

AIの利点



1

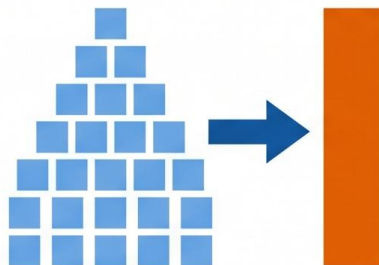
24時間365日稼働



休まず連続して動作

2

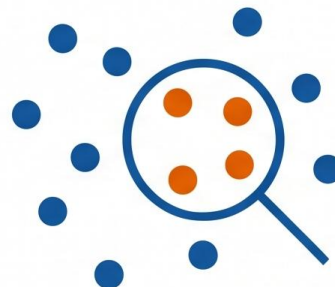
大量データを
高速処理



膨大な情報を短時間で

3

細かいパターン
を検出



人が見落としがちな
パターンを検出

4

反復作業を
効率化



繰り返し作業を自動化

**自律的な
創造性・
柔軟性の限界**

ゼロから新しい
概念を生み出す
ことは苦手

**予期せぬ
状況に弱い**

想定外の入力や
前例のない事態
の対応が不得手

**倫理的判断
には
人間が必要**

善悪や価値の判断
が伴う場面では人
間の介在が不可欠

**学習データ
の質・偏
に依存**

偏ったデータでは
結果の精度が
低下する

欠点を理解して活用する



2-3 AI の種類

機械学習

コンピュータがデータを用いて自ら学習し、知的能力を向上させる技術。データを与えて学習させることで、より正確に予測・分析・合成ができる。

柔軟性が高い / 説明性は低い

両者の違い

データから学習
⇔ルールで判断

柔軟だが不透明
⇔明確だが硬直的

知的なITシステム

人間が書いたルールや知識を用いて、意思決定や問題解決を行う人工知能。
人間が直接プログラムする。

説明可能性が高い / 柔軟性は低い

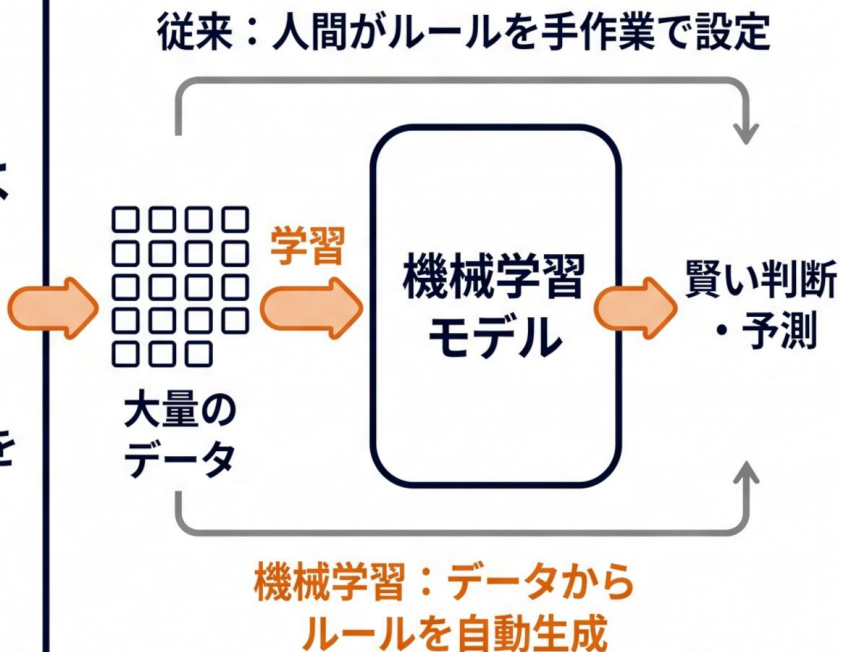
両者の欠点と利点を理解したうえで、最終判断は人間が行う

AI = 道具 人間 = 判断と責任

機械学習の定義と特徴

機械学習とは

コンピュータがデータを使って学習し、知的能力を向上させる技術



情報の抽出

データからパターンや関係性を自動で発見

簡潔さ

人間が設定していたルールを自動で生成

限界の超越

従来の方法では困難だった課題も解決できる可能性



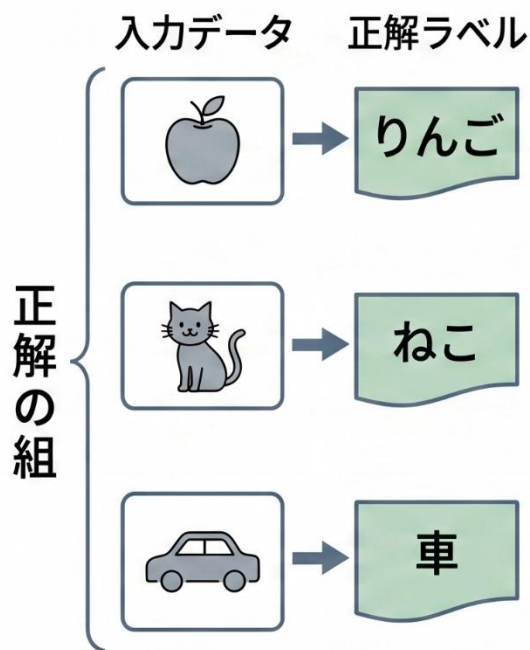
機械学習の「教師あり学習」の仕組み



正解の組を用いる方式を 教師あり学習 と呼ぶ。

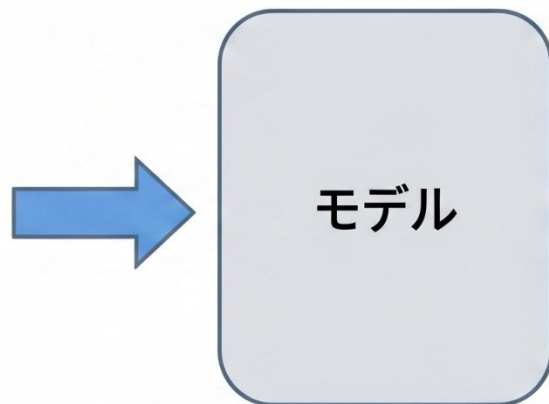
(このほかに、正解なしでデータの構造を見つける教師なし学習や、試行錯誤を通じて学ぶ強化学習といった方式もある。)

1. 大量の訓練データ



2. 学習プロセス

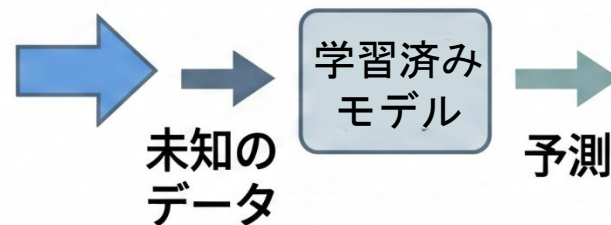
大量の訓練データを用いて学習を行う。



正解の組を用いる方式を教師あり学習と呼ぶ。

3. 学習済みモデル

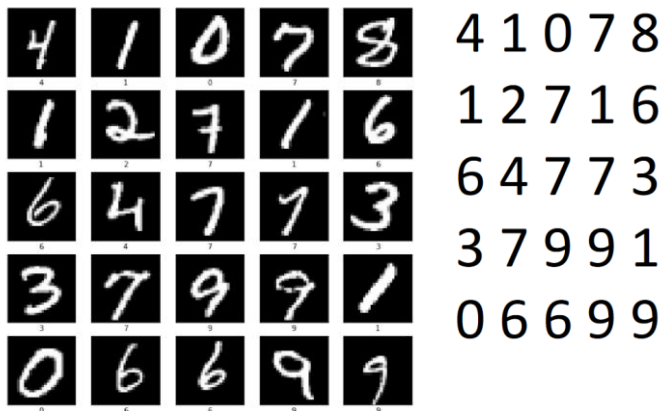
未知のデータから予測が可能になる。



機械学習の訓練データと学習の仕組み



大量の訓練データを用いて
学習を行う



例：手書き数字の
画像60000枚

それぞれの正解
(4, 1, 0, 7, 8, ...)
60000個

⇒訓練データとして使用する。

機械学習のプログラムが訓練
データを用いて学習を行う

プログラム

```
[4] !pip install -U scikit-learn matplotlib
import torch
import torch.nn as nn
import torch.optim as optim
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# データの取得と前処理
iris = datasets.load_iris()
X = iris.data
y = iris.target

# データの標準化
scaler = StandardScaler()
X = scaler.fit_transform(X)

# 訓練データとテストデータの分割
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

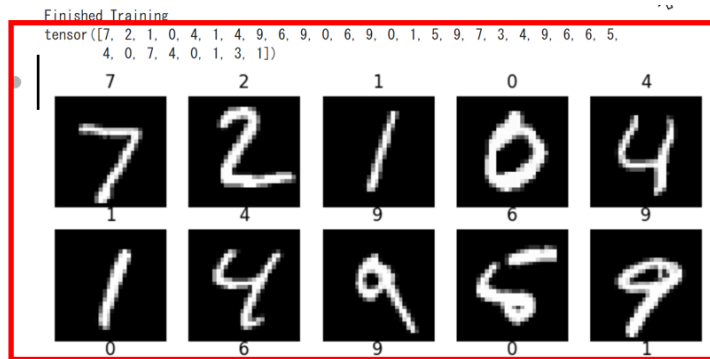
X_train = torch.tensor(X_train, dtype=torch.float32)
y_train = torch.tensor(y_train, dtype=torch.long)
X_test = torch.tensor(X_test, dtype=torch.float32)
y_test = torch.tensor(y_test, dtype=torch.long)

# ニューラルネットワークの定義
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(4, 10) # 入力4次元 (Irisの特徴量)
        self.fc2 = nn.Linear(10, 3) # 出力は3クラス

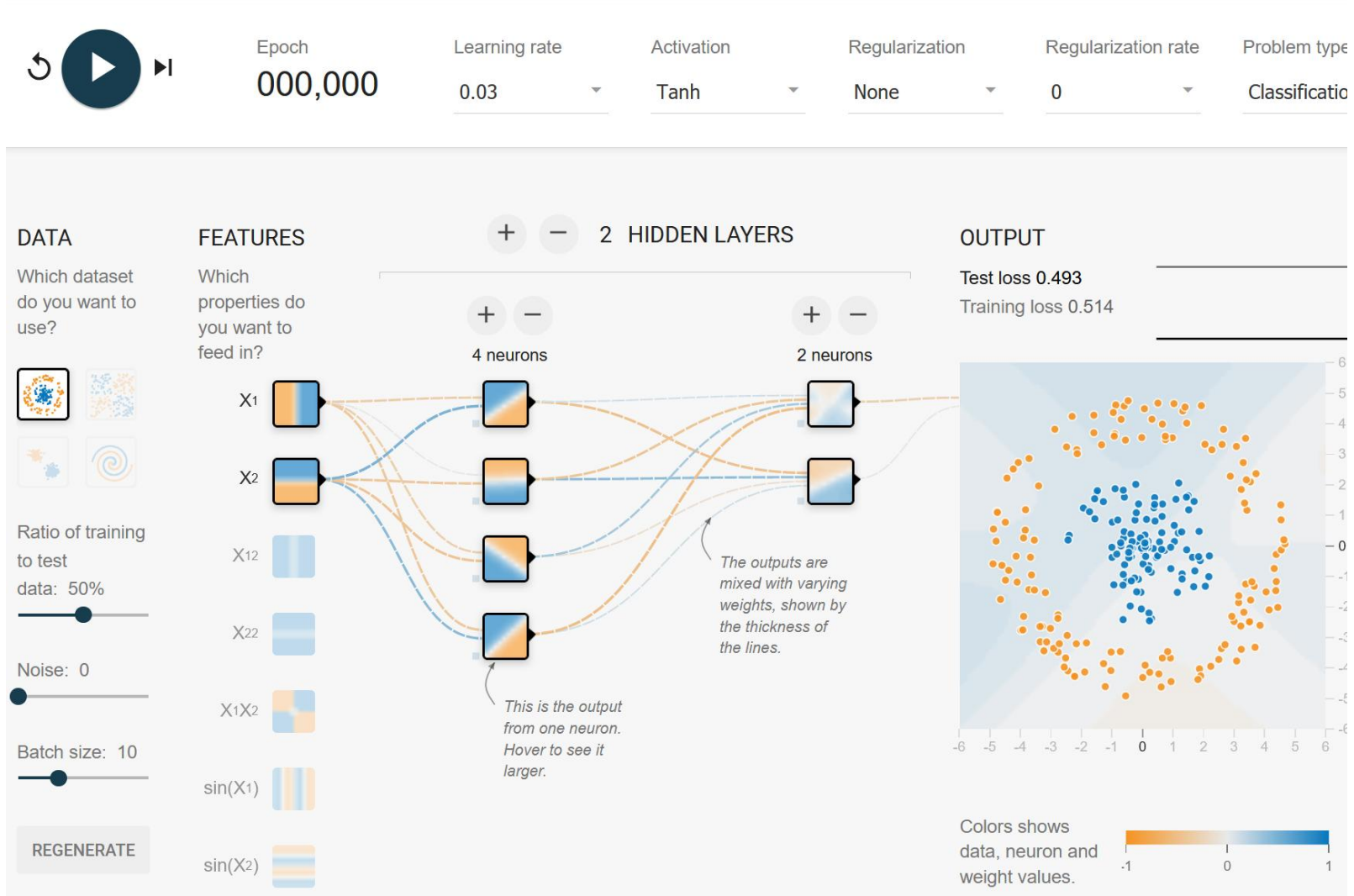
    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = self.fc2(x)
        return x

net = Net()
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(net.parameters(), lr=0.01)
```

学習の結果、文字認識の能力を獲得する。
学習ののち、新たな手書き数字の画像分類
を行うことができる。



ニューラルネットワークについて学ぶためのサイト TensorFlow Playground



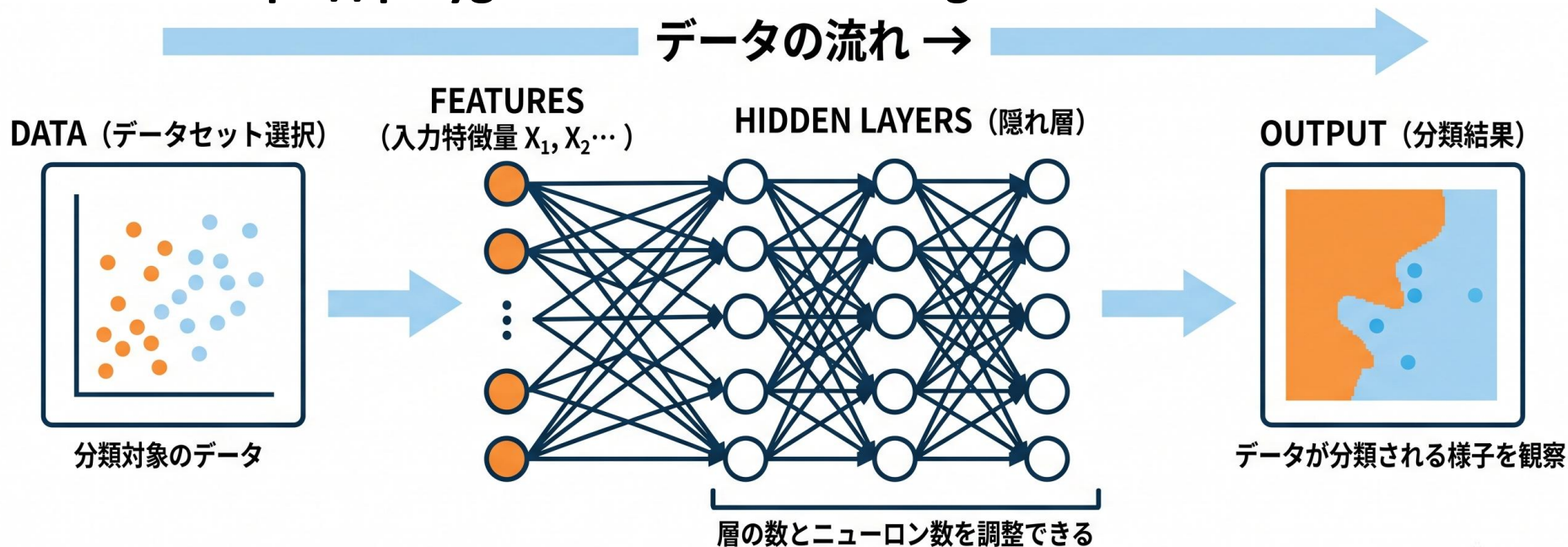
<https://playground.tensorflow.org>

ニューラルネットワークについて学ぶためのサイト TensorFlow Playground



ニューラルネットワーク：訓練データを与えて学習させる機械学習の代表的手法

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



学習パラメータ (学習中に調整する設定)

Learning rate (学習率)：学習の進み方の速さ

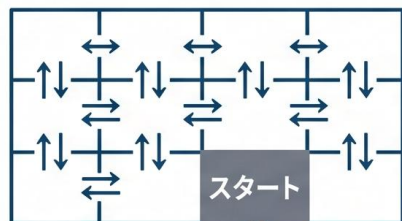
Activation (活性化関数)：信号の変換方法

Epoch (学習回数)：データを繰り返し学習

人間が書いたルールや知識を用いて、意思決定や問題解決を行う人工知能。

探索による問題解決

すべての組み合わせを試して
答えを見つける



1手目

2手目

3手目

到達地点

3回動いた後の行き先を、コンピュータが
全経路を探索して出力する

エキスパートシステム

専門家の知識を「もし～ならば～」
のルールにして判断させる

専門家の知識

もし [条件A]
ならば [結論X]

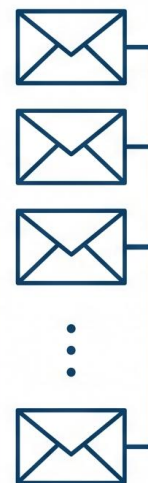
もし [条件B]
ならば [結論Y]

もし [条件B]
ならば [結論Y]

コンピュータ
が判断

ルールベースのフィルタリング

あらかじめ定めた条件で
データを自動分類する



機械学習と知的なITシステム



機械学習



知的なITシステム



知識の由来

ルール

人が記述

データから自動で獲得

強の由来

人間が明示的に記述

手書き数字認識、画像生成

具体例

エキスパートシステム、スパムフィルタ

人間が言語化しにくいパターンにも対応

強み

判断根拠を人間が説明しやすい

なぜその結果になったか説明しにくい場合がある

弱み

ルールの追加・修正に人手がかかる

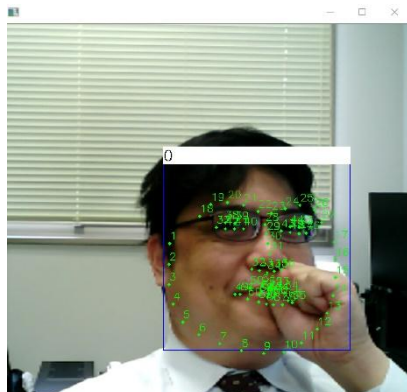
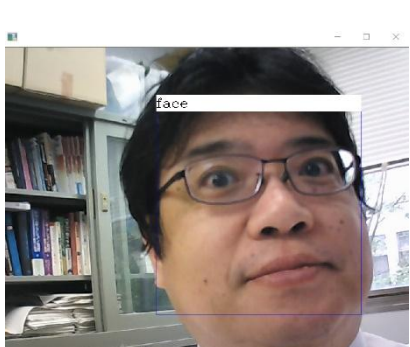
両者は相反するものではなく、実際のシステムでは組み合わせて使われることもある

2-4 AI でできること (応用分野)

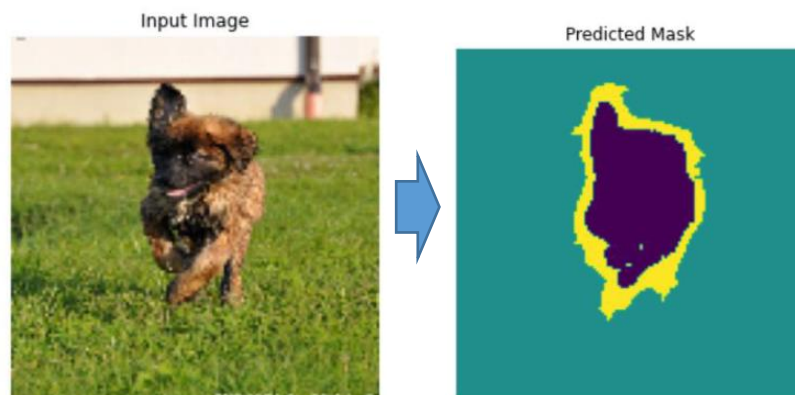
人工知能の応用例



人工知能は私たちの生活や仕事を大きく変革する技術。



顔検知、顔識別



画像のセグメンテーション



+



→



金子の顔

有名人の声、表情、語り

金子がその有名人そっくりで語りだす

合成



This is an implementation of [Mask R-CNN](#) on Python 3, Keras, and TensorFlow. The model generates bounding boxes and segmentation masks for each instance of an object in the image. It's based on Feature Pyramid Network (FPN) and a ResNet101 backbone.

Webブラウザで翻訳を行う
Mate Translate (Web ブラウザ
Firefox のアドオン)

人間の言葉、声の処理：対話型AI（チャットボット）



- 文章の翻訳，要約，推敲のほか，相談や自由なアイデア出しに使用できる。
- ユーザーが質問や依頼を入力すると，AIが回答やコードを生成。

KU Java うるう年の計算法 分からない

Javaでうるう年の計算方法は、以下のようになります。

```
if (year % 4 == 0) {
    if (year % 100 == 0) {
        if (year % 400 == 0) {
            // うるう年
        } else {
            // 平年
        }
    } else {
        // うるう年
    }
} else {
    // 平年
}
```

上のJavaのコードは、年が4で割り切れる場合はうるう年の可能性があります、その後100で割り切れる場合は平年となります。それでもうまくいかない場合には、400で割り切れる場合はうるう年となります。

KU 曜日の計算はどうなりますか

曜日の計算には、日付から曜日を求めるためのアルゴリズムがいくつか存在します。その中でも一般的に使われるアルゴリズムに、「Zellerの公式」と呼ばれるものがあります。

Zellerの公式は、以下の式で日曜日から土曜日までの7日間の曜日を求めることができます。

```
w = (d + floor(2.6 * m - 0.2) + y + floor(y / 4) + floor(c / 4) - 2 * c) mod 7
```

ただし、

- w: 曜日 (0:日曜日, 1:月曜日, ..., 6:土曜日)
- d: 日
- m: 月(3月から12月を1月から8月に変換)
- y: 年(世紀から19世紀)
- c: 世紀

例えば、2022年3月1日の曜日を求める場合、以下のように計算します。

```
w = (1 + floor(2.62 - 0.2) + 22 + floor(22/4) + floor(20/4) - 220) mod 7
w = (1 + 2 + 22 + 5 + 5 - 40) mod 7
w = (35) mod 7
w = 1
```

結果的にw = 1となり月曜日になります。

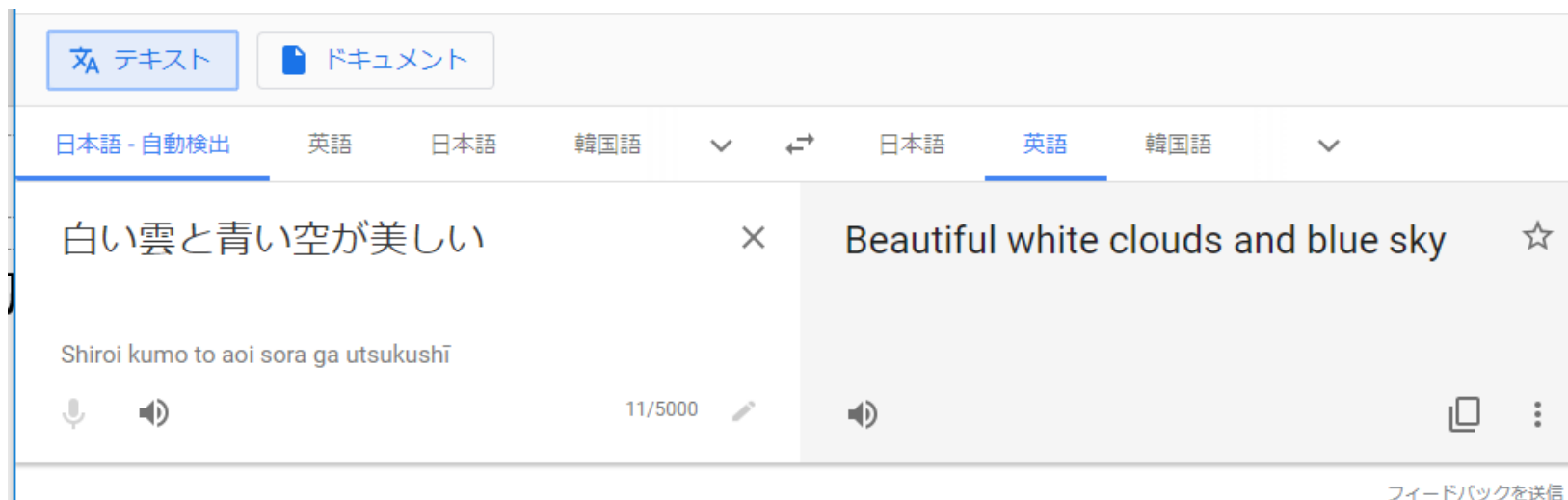
Javaで実装する場合には、上記の式をそのまま記述することもできますが、Javaには日付や時間に関連するクラスが用意されており、それらを使用することも簡単に曜日を求めることができます。

人間の言葉、声の処理：自動翻訳サービス



Webブラウザ上でテキストを入力すると，別の言語に自動翻訳される。

例：DeepL, Mate Translate (Webブラウザ Firefox のアドオン)



DeepL の URL: <https://www.deepl.com/ja/translator>

視覚情報処理：画像分類



画像を入力すると，AIがその内容を認識し，ラベル（Forehead 98%，Glasses 98%，Chin 97% 等）を付けて分類。

Faces Objects **Labels** Properties Safe Search

The screenshot shows the 'Labels' tab of the Google Cloud Vision API interface. On the left, there are two side-by-side images of a man with glasses and a white shirt. The right image is labeled '126.png'. On the right side, a list of labels with their corresponding confidence percentages is displayed, each with a green progress bar:

Label	Confidence
Forehead	98%
Glasses	98%
Chin	97%
Eyebrow	95%
Vision Care	94%
Dress Shirt	90%
Jaw	88%
Smile	88%

元画像

画像分類の結果

Google Cloud Visionのオンラインサービス

URL： <https://cloud.google.com/vision/docs/drag-and-drop>

視覚情報処理：オブジェクトの発見・検知



画像の中のオブジェクト（人，自転車，車など）を，AIが
発見・検知する。

（例）DeepLabv3+を使用した例では，街路の画像から人，
自転車，車，道路，建物等を色分けして検出。



元画像

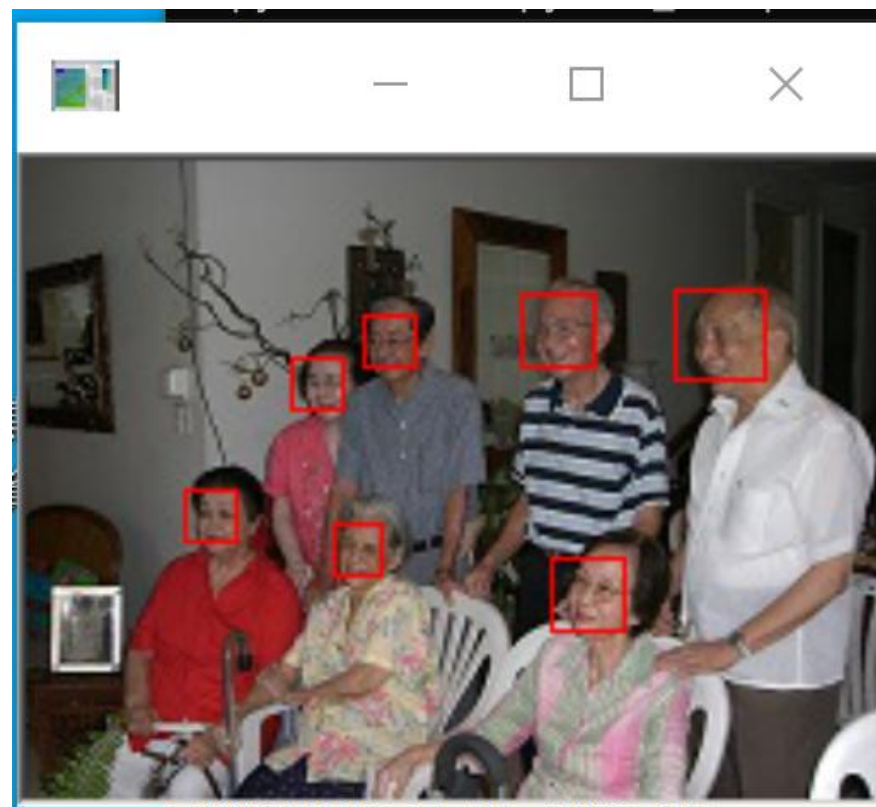
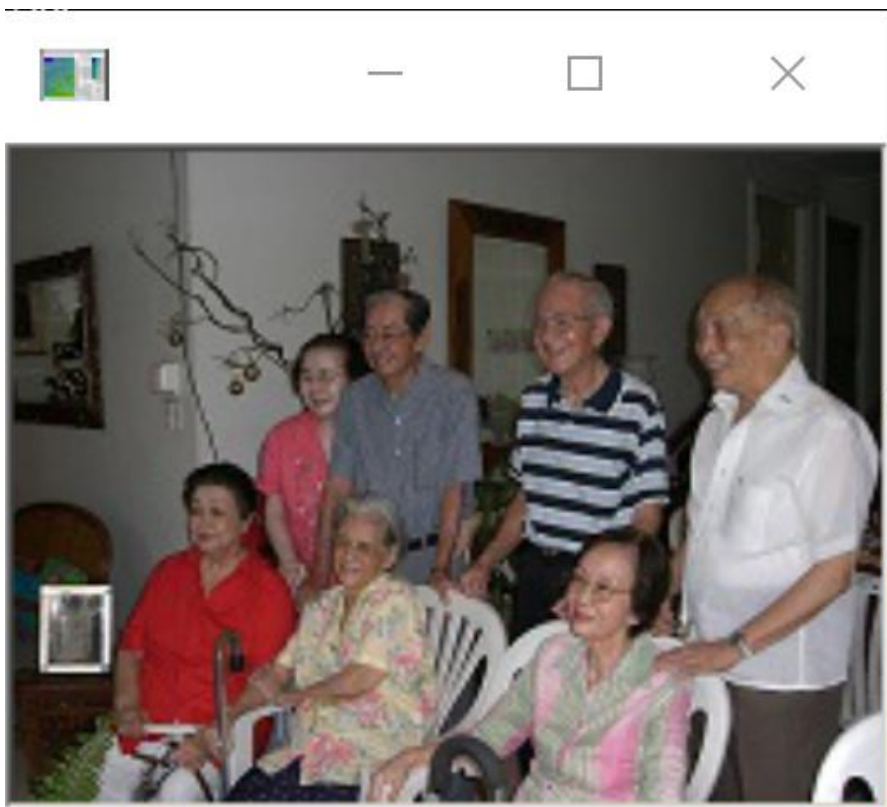


人工知能による読み取り結果
(DeepLabv3+ を使用)

視覚情報処理：顔検知

画像中の**顔の位置と大きさの情報**を読み取る技術。

集合写真のような複数人が写る画像からも，各人の顔を矩形で囲んで検出できる。

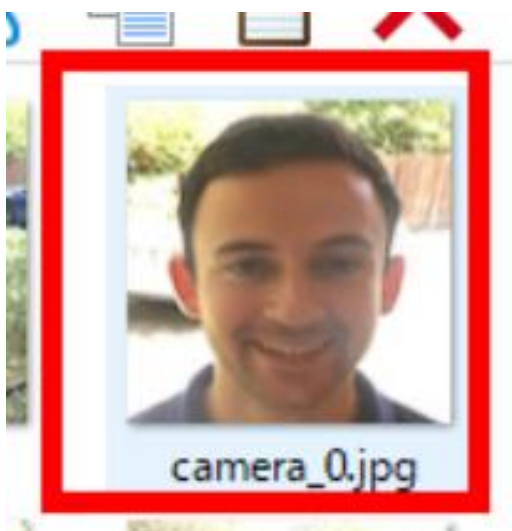


視覚情報処理：顔識別（顔からの人物特定）



あらかじめ登録された顔データとの距離（類似度）を計算し、最も距離が近い人物を特定する技術。

（例）入力画像に対して各人物との距離を算出し、最小距離の人物名（younes, distance 0.309...）を出力する。



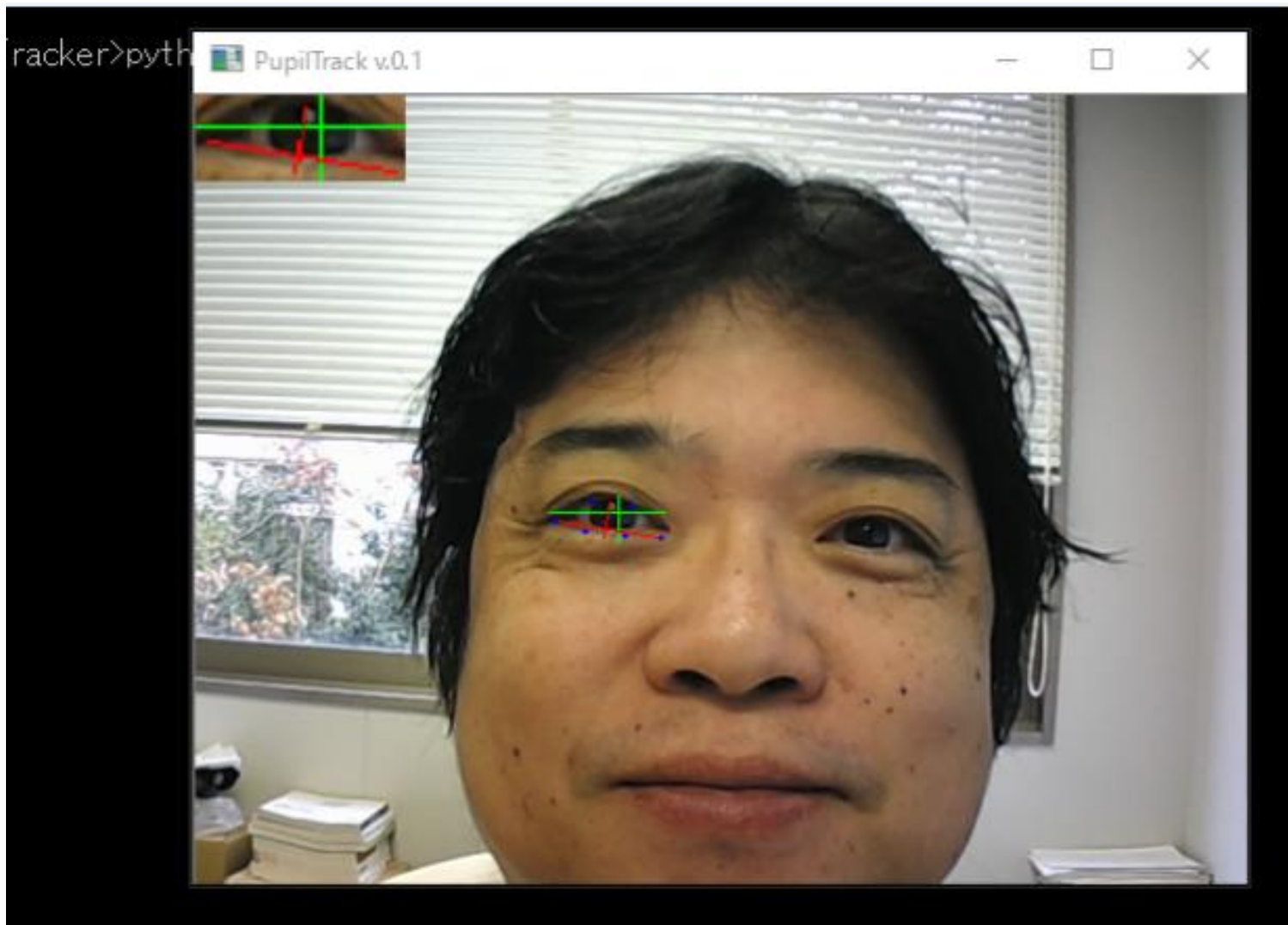
```
--for danielle, the distance is 0.4635717
--for younes, the distance is 0.30962762
--for tian, the distance is 0.48845953
--for andrew, the distance is 1.0392754
--for kian, the distance is 0.8913959
--for dan, the distance is 0.551507
--for sebastiano, the distance is 0.45932084
--for bertrand, the distance is 1.0153409
--for kevin, the distance is 0.80856085
--for felix, the distance is 0.7121804
--for benoit, the distance is 0.39749846
--for arnaud, the distance is 0.7137512
it's younes, the distance is 0.30962762
```

younes

視覚情報処理：目の動きの読み取り



瞳孔の位置と動きを追跡する技術.



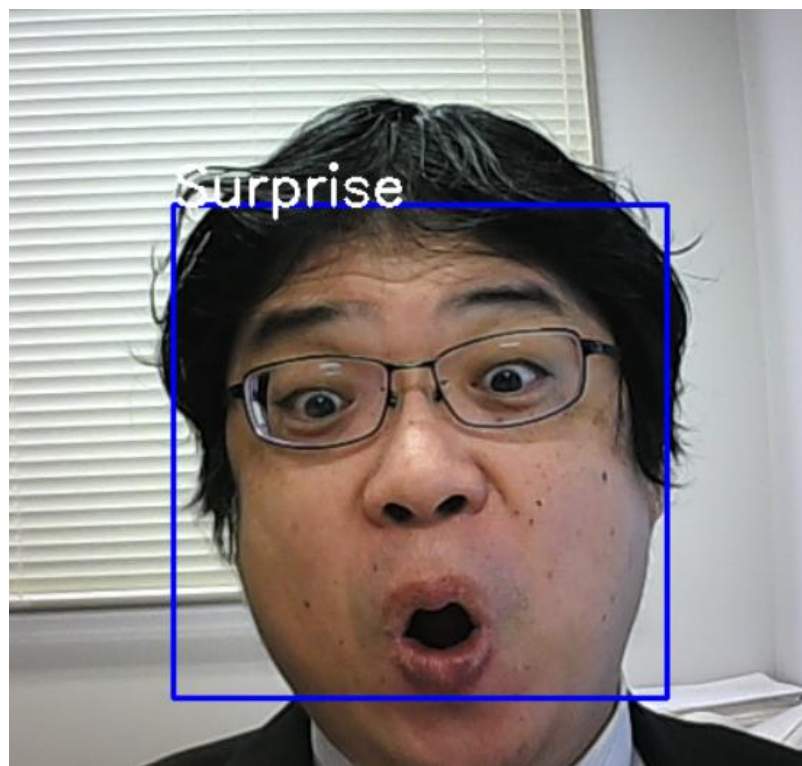
目の動きの読み取り（Pupil Tracker を使用）

視覚情報処理：表情の自動判定



顔画像から表情を自動で判定する技術.

(例) Angry, Disgust, Fear, Happy, Neutral, Sad, Surprised
の各カテゴリの確率を算出. Surprised 52.88% と判定され
た場合, 「驚き (Surprised)」 と判定される.



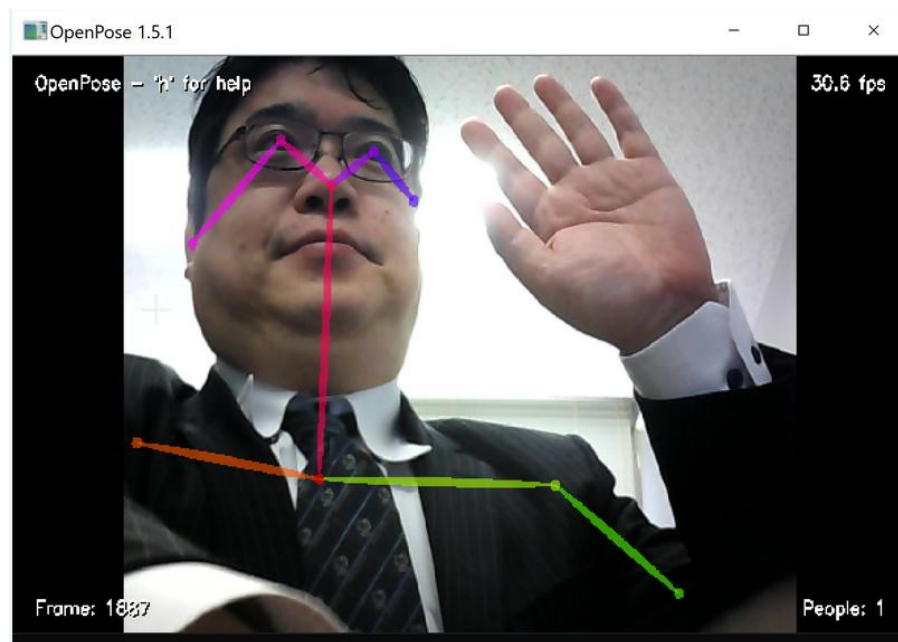
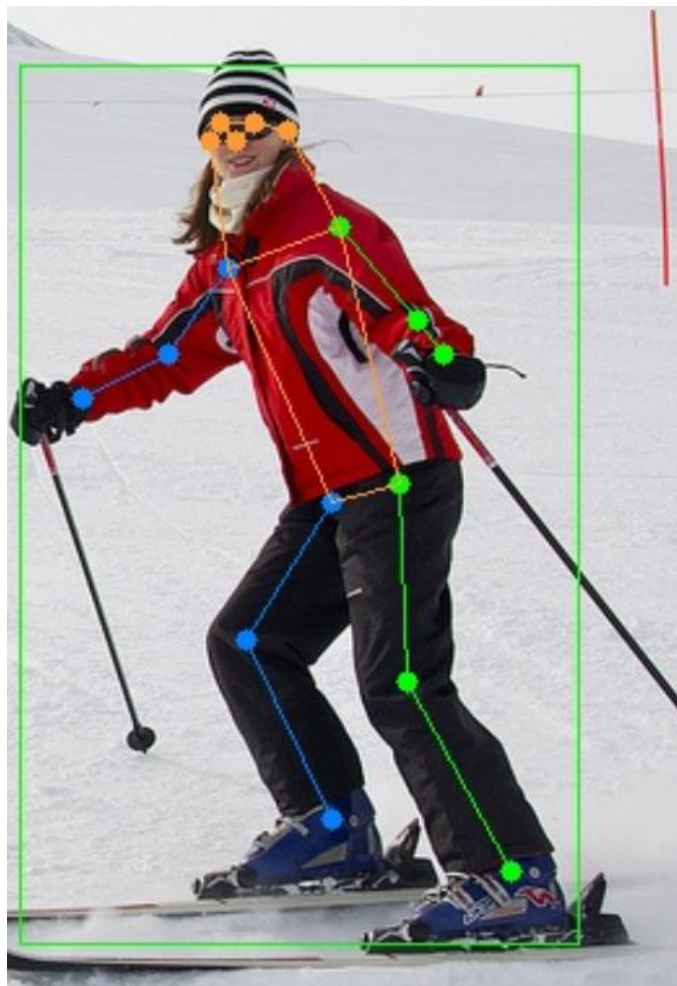
```
Angry: % 4.94537390768528  
Disgust: % 7.72874653339386  
Fear: % 2.0912714302539825  
Happy: % 1.1880283243954182  
Neutral: % 30.127882957458496  
Sad: % 1.0293880477547646  
Surprised: % 52.88930535316467
```

「驚き (Surprised)」 と判定
されている

視覚情報処理：人体の姿勢の読み取り

人体の関節位置を検出し，骨格を線で結んで姿勢を可視化する技術。

(例) スキーや野球などの動きの解析にも応用される。



人体の姿勢を読み取り
(OpenPose を使用)





視覚情報処理：群衆の数のカウント

群衆の画像から人数を自動で数える技術。各人の位置を矩形で囲んで検出する。



元画像



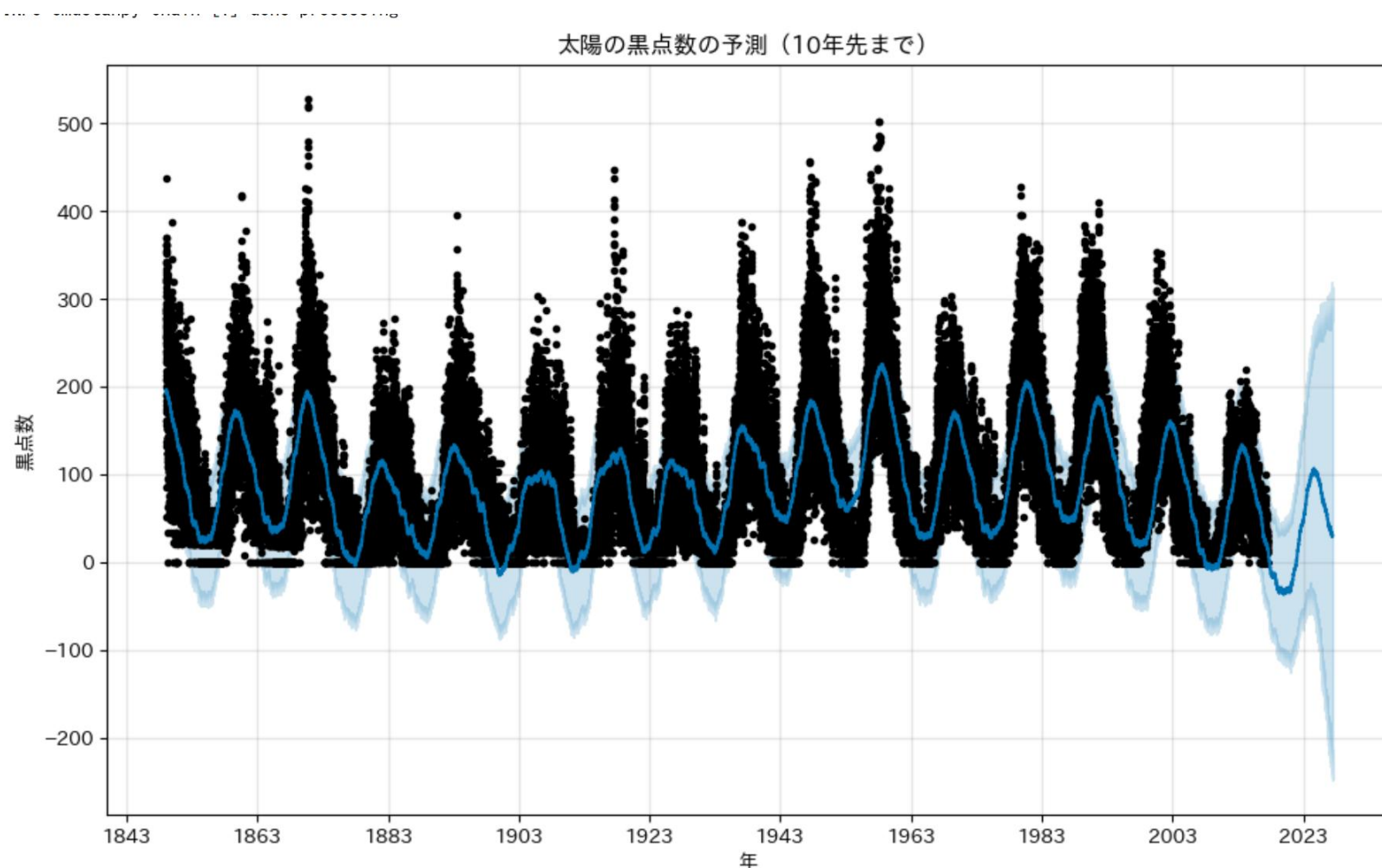
群衆の数のカウント
(FIDTM を使用)

データ分析と予測

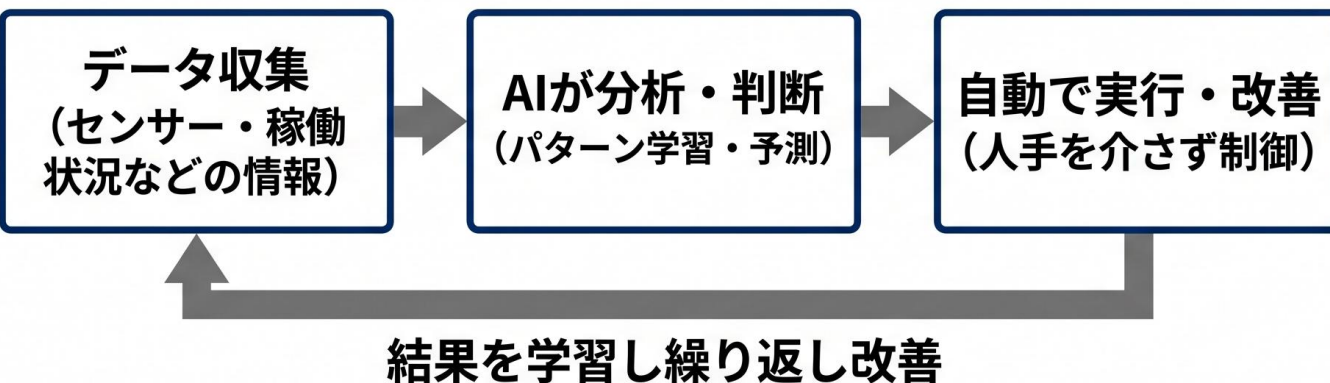


過去のデータを基に将来予測を行う技術。

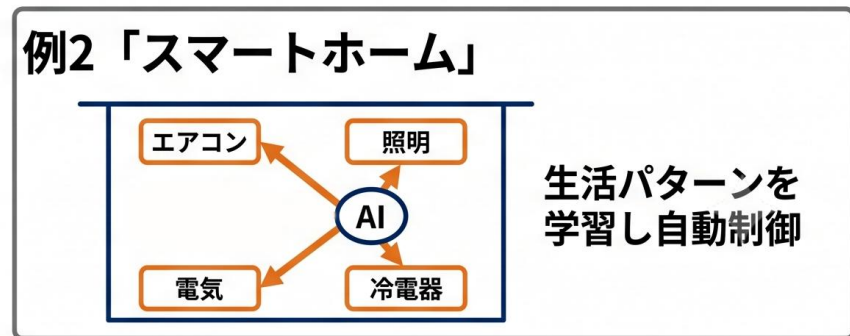
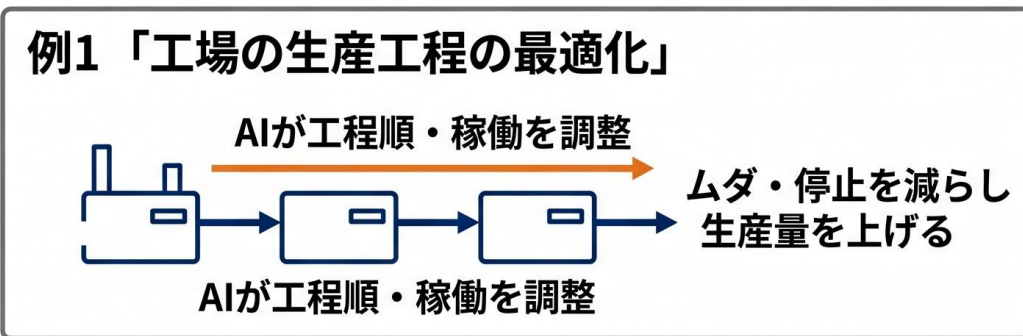
(例) 過去の太陽の黒点数から将来の傾向を予測



作業やプロセスをAIで自動化・最適化



- A** 自動化：人の作業を機械が代行
- +
- B** 最適化：最も効率の良い方法を選ぶ



人工知能と知識表現：Prolog による推論



コンピュータが、知識から推論して答えを導く

①知識をルールで書く

```
human(hanako). 『hanakoは人間』  
human(taro). 『taroは人間』  
think(X) :- human(X).
```

ならば (条件)

『Xが人間なら、Xは考える』
というルール

②質問する

```
human(X).
```

『Xは人間か?』
と尋ねる

③推論して答えを得る

```
human(hanako)  
human(taro)
```

```
X = hanako  
X = taro
```

あてはまるものを
すべて返す

事実とルールを照らし合わせ、条件を満たす答えを自動で見つけ出す = 推論

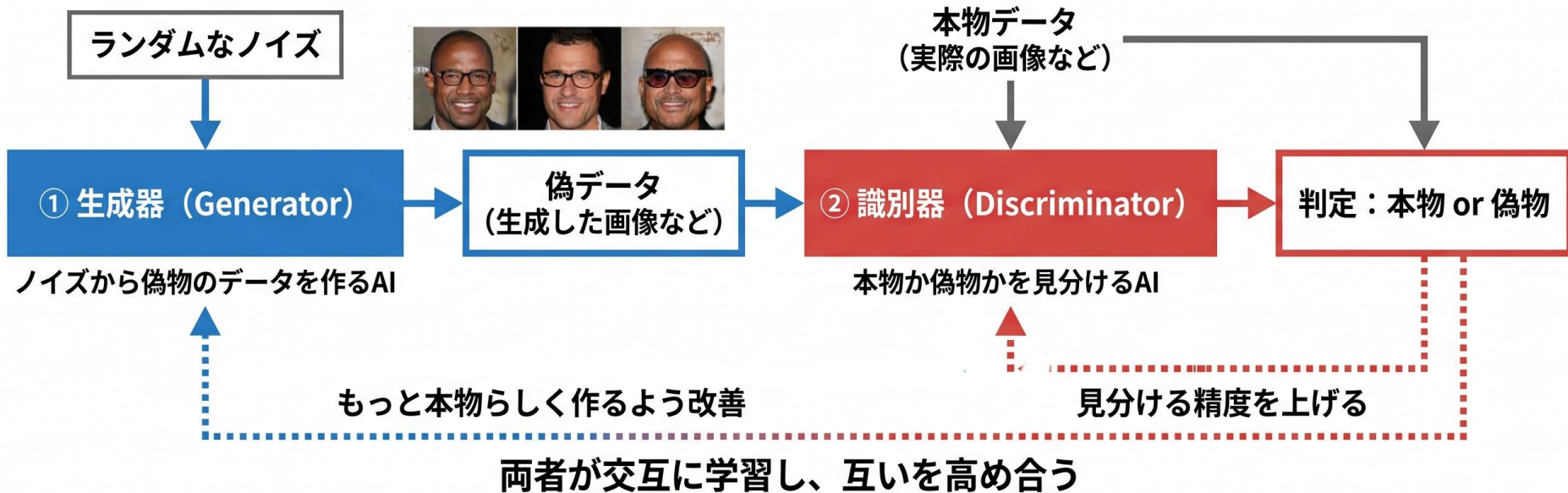
2-5 AI による合成

GAN（敵対的生成ネットワーク）：2つのAIが競い合って学習する仕組み



AIによるデータの生成能力を示す研究のひとつ。

GANの学習の仕組み



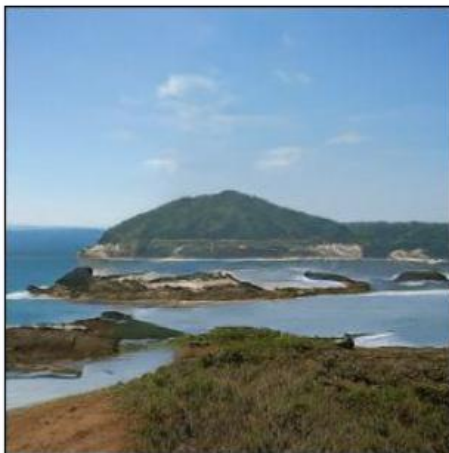
たとえば：生成器＝精巧な「偽札作り」、識別器＝偽札を見抜く「鑑定士」。
偽札作りは見破られないよう腕を上げ、鑑定士は見抜く目を鍛える。
この競争のくり返して、生成器は本物そっくりのデータを作れるようになる。

GANの応用例

さまざまな種類の画像（犬，風景，蝶，食べ物等）を，安定して高精細に生成

Large Scale GAN Training for High Fidelity Natural Image Synthesis

[Andrew Brock](#), [Jeff Donahue](#), [Karen Simonyan](#), <https://arxiv.org/abs/1809.11096v2>



GANと現在の画像合成（Text-to-Image）の関係



GAN (2014年)

生成器 (Generator)



競い合って学習

識別器
(Discriminator)

AIがリアルなデータを生成できることを示した重要な研究

生成AIという分野の誕生

GANが生成AIの扉を開いた

その後、新たな生成手法が次々と登場

拡散モデル (Diffusion Model)



ノイズ



ノイズから徐々に画像へ



画像

現在のText-to-Image (テキスト画像合成) の基盤

Text-to-Image (テキストから画像を生成) = 次ページへ

テキストからの画像合成

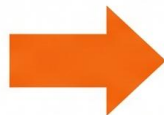


テキスト（文章）の指示に基づいて、AIが写真やイラストを人工知能が合成する技術

テキスト指示（入力）

夕暮れの海辺を
歩く犬の写真

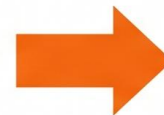
言葉の意味
を理解



AIによる変換

- 指示文を解析
- 対象・情景・画風を把握
- 画像として組み立て

画像を生成



画像（出力）



主なサービス例とさらなる発展

HappyHorse（無料試用・登録必要）

Microsoft 365 Copilot

動画生成・編集機能へと発展

HappyHorse の URL: <https://www.happyhorse.com/>



テキスト「森の中で本を読んでいる柴犬のイラスト。暖かい午後の光が差し込んでいる。」からの画像生成例（Copilotを使用）



営業部

テキスト「「営業部」と書かれた木製のドアプレート。シンプルで清潔感のあるデザイン。」からの画像生成例（Copilotを使用）



テキストからの動画生成例（Gemini Veo 3.1を使用）



入力画像



結果

画像とテキスト「添付の画像の人物が爆笑している。背景はハワイの海岸。」からの画像生成例（Copilotを使用）※画像をクリップボードにコピーして、Copilotに与えた



入力画像



結果

画像とテキストからの動画生成例（Happy Horse を使用）



入力画像
絵画モナリザ



画像とテキストからの動画生成例（Happy Horse を使用）

写真とビデオから合成を行うAI



【入力】

【AI処理】

【出力】

入力1：1枚の顔写真（人物A）

人物A

動かしたい人物の見た目

入力2：演技ビデオ（人物B）

人物B

声・表情・口の動き・語り

①人物Aの顔の特徴を抽出

②人物Bの動き・音声を解析

③Aの顔にBの動き・声を合成

合成ビデオ



写真の人物Aが、人物Bの声・表情・語りですっきりと話し出す映像

= 見た目はA、動き・声はB。両者を1本の映像に融合する技術



写真

+



ビデオ

→



人工知能により
合成されたビデオ



2-6 AIの現状と注意点

① AI の利用での注意点



■ AIの限界

AIは誤った結果を出すことがある

AIができることは限定的。
すべてを任せてはいけない

■ AIは人間が決めた範囲の中だけで動く
範囲外のことは判断できない

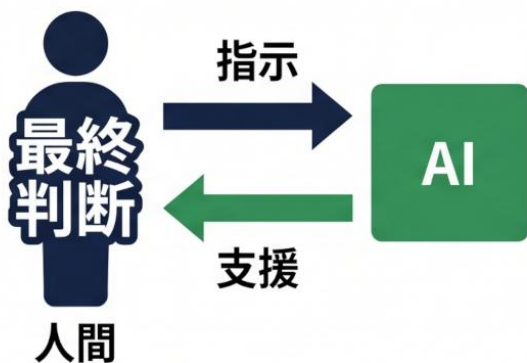


■ AIは人間が使う道具

AIを使うかどうか、どう使うかは人間しだい
人間の判断や考えを補う (補完する) ために使う

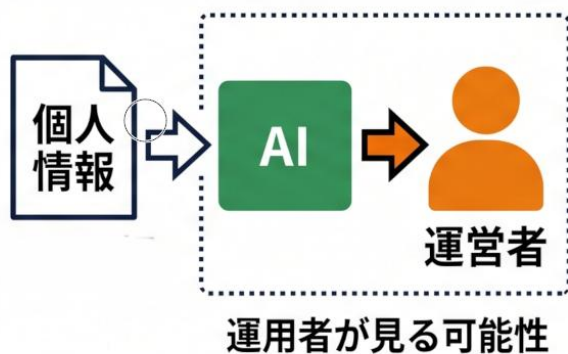
② AI を責任をもって利用するために

人間が主導



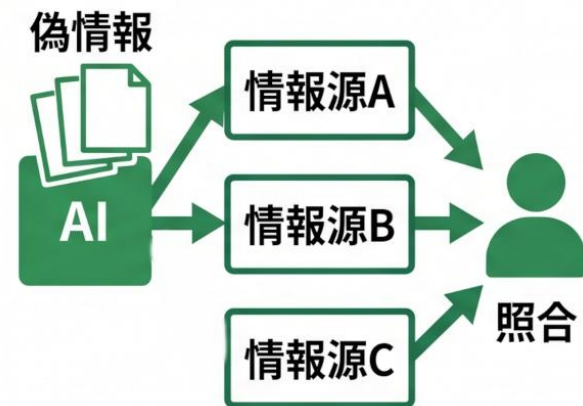
AIは人間を支援する道具。
最終的な判断は人間が行う。

個人情報の取り扱い



危険なのはAIより運営者の
場合がある。
個人情報を与える際は注意。

偽情報の抑制



AIで偽情報の生成が容易に。
複数の信頼できる情報源を
照合する習慣を。

人工知能との向き合い方



まず「自分で調べ、考える」

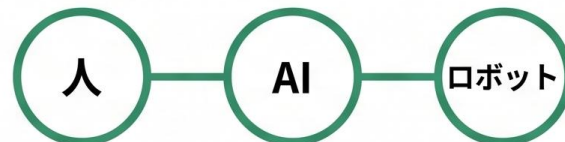
高度に発達した人工知能が脅威になるかどうかは、他人の意見をうのみにせず、自分で調べ、根拠をもって考察することが重要。

人間のクリエイティビティを高める

人がより創造的になるための道具として人工知能を活用する。

人・AI・ロボットの協働

人間と人工知能とロボットが協力して働くことが、今後必然的に起こりうる。



演習

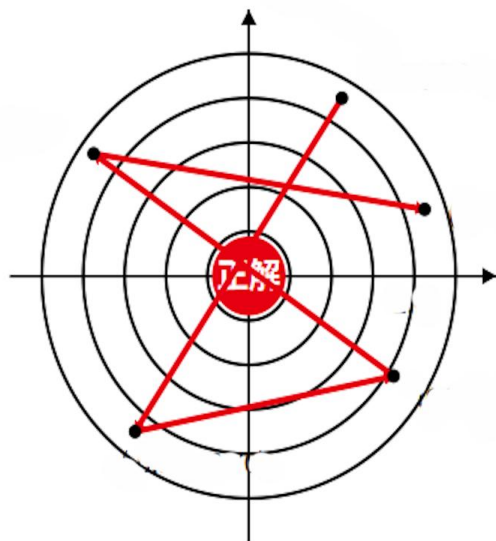


演習の前に：「学習率」とは



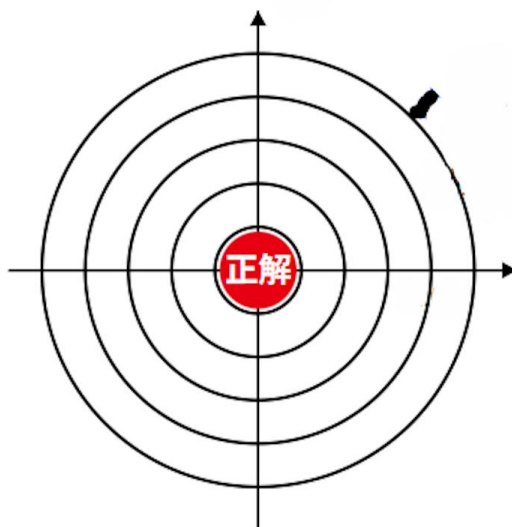
学習率は「1回の学習でどれだけ大きく修正するか」を決める値

学習率が大きすぎる



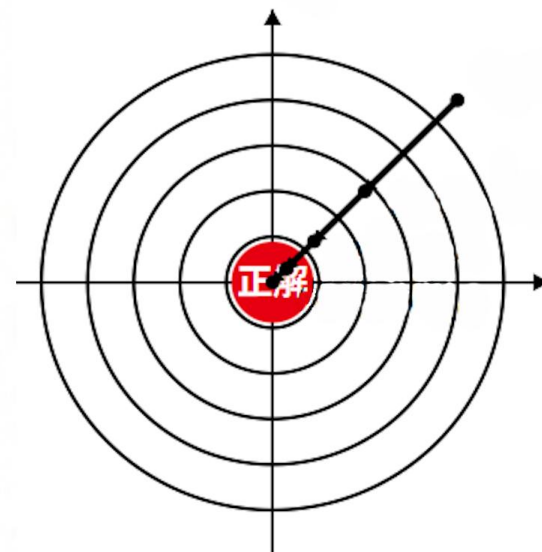
修正が大きすぎて正解を飛び越え、学習が不安定。

学習率が小さすぎる



少しずつしか修正せず、学習に多くの時間がかかる。

適切な学習率



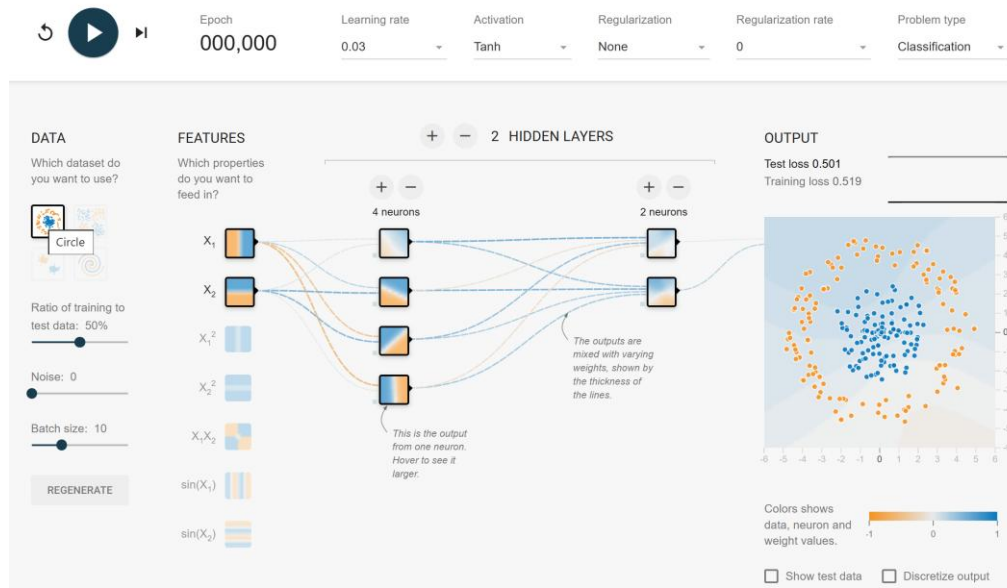
安定して正解に近づいていく。

演習 1 のTensorFlow Playgroundで学習率を0.001, 0.03, 1などに
変えて、学習の安定性がどう変化するか観察してみよう

1. ニューラルネットワークの学習過程の可視化



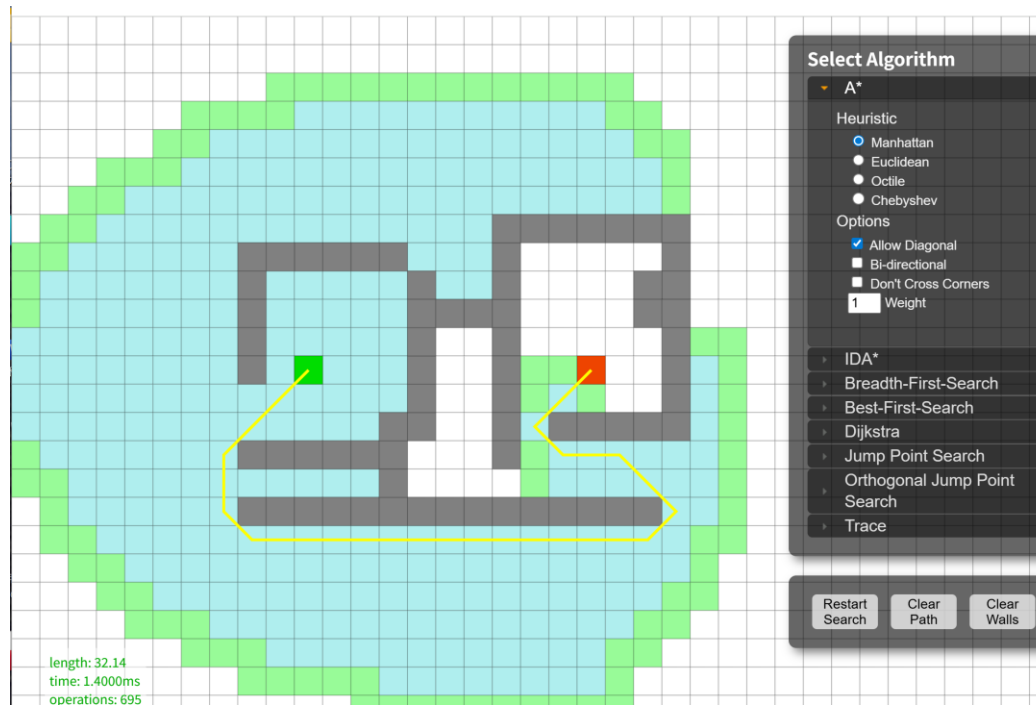
- ニューラルネットワークの構造やパラメータを変えながら、学習の過程を観察できる。
- TensorFlow Playgroundを使用。
- 手順：**データセットを選ぶ** → **隠れ層の数やニューロン数を設定する** → **再生ボタンで学習を開始** → **分類結果の変化を観察する**。
- URL: <https://playground.tensorflow.org>



2. 探索アルゴリズムによる経路探索の可視化



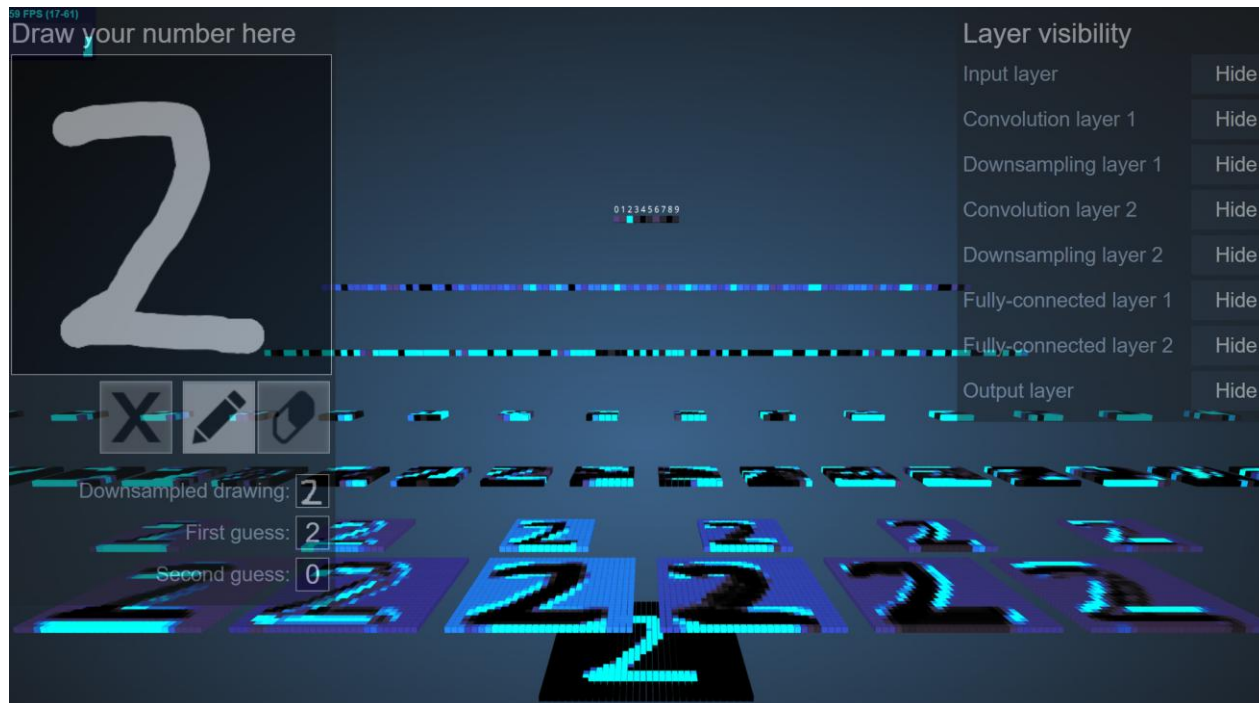
- 探索アルゴリズムで、スタート地点からゴール地点までの経路を見つける過程を可視化
- 手順：グリッド上に壁を配置する → アルゴリズムを選択する → 探索を開始 → 探索の広がり方と最短経路を観察する。
- URL: <https://qiao.github.io/PathFinding.js/visual/>



3. 畳み込みニューラルネットワークの内部構造の可視化



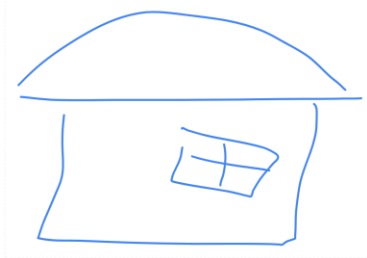
- 畳み込みニューラルネットワーク（CNN）の各層が入力画像をどのように処理しているかを，3Dで可視化.
- 手順：描画パッドに数字を描く → 各層のニューロンの活性化がリアルタイムに表示される → ニューロンにカーソルを合わせて検出内容を確認する.
- URL: https://adamharley.com/nn_vis/cnn/3d.html



4. 人間の下書きを人工知能が清書する



人工知能が、元の情報を保ったまま人間のイラストを清書する技術



人間がイラストを描く

コンピュータが候補を出す

完成

AutoDraw のURL: <https://www.autodraw.com/>