

# mi-6. 人工知能でできること, プロダクションシステムでの知識表現と 推論エンジン

## 人工知能を演習で学ぶシリーズ (6)

<https://www.kkaneko.jp/dblab/intro/m/index.html>

金子邦彦



# 6-1 人工知能デモサイト

# 人工知能

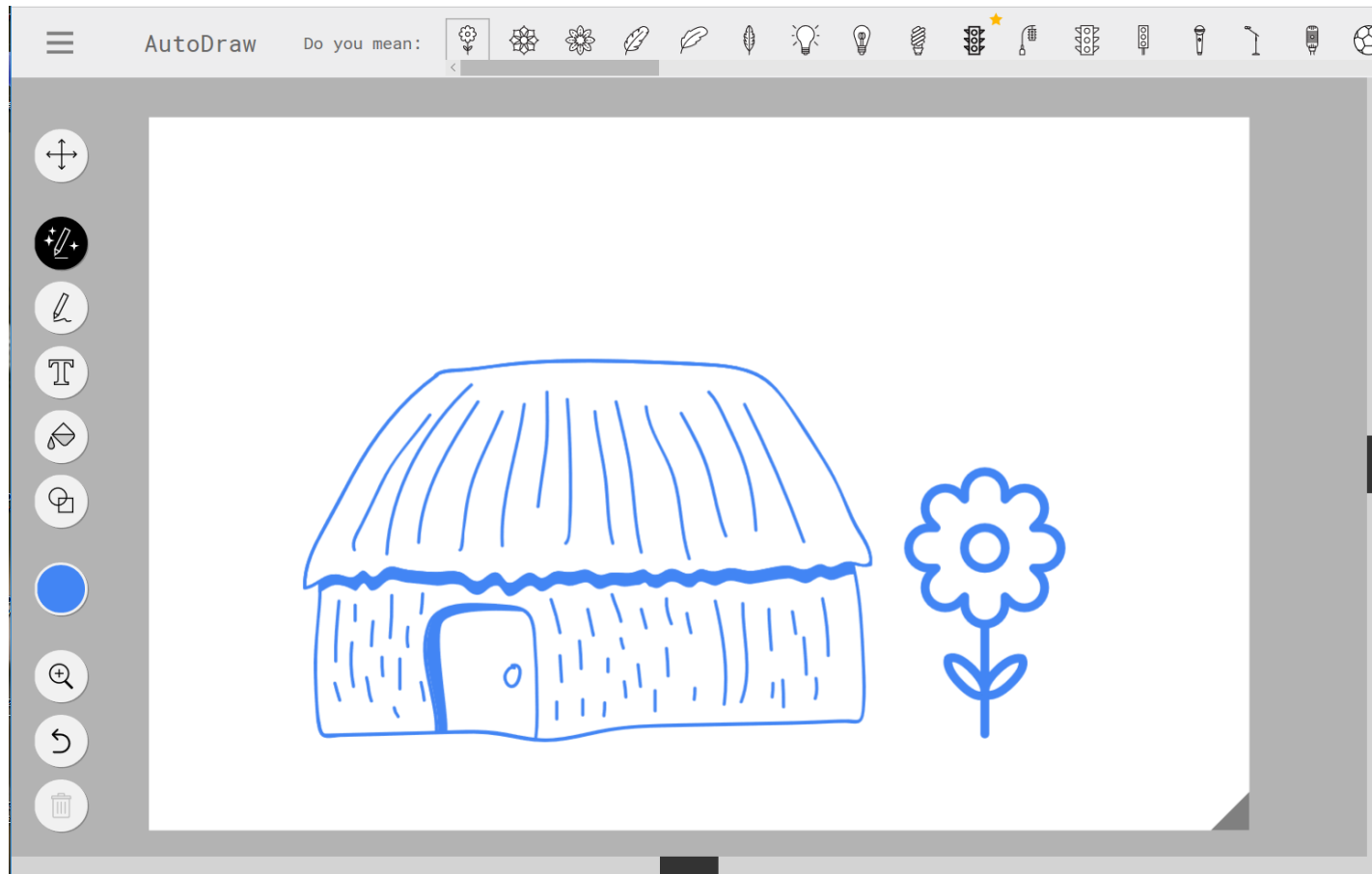
**人工知能 (AI)** とは、  
**機械が、人間の知性を示すこと**

- ・ 重要な学問領域である
- ・ 「コンピュータに人間の仕事を**代行させた**い」と広い意味で使うことも

# いまの人工知能 (AI) で、できること

- 音声認識 (人の声を「文字」化する)
- 画像解析 (画像の中から、「もの」を見つける)
- 創作
- 欠損の補充
- 予測
- 自動運転車
- 翻訳

# 人工知能を使ってイラストを描く AutoDraw



# 人工知能を使ってイラストを描く AutoDraw

① ウェブブラウザで次の URL を開く

<https://www.autodraw.com/>

② 「**Start Drawing**」 をクリック



AutoDraw

Fast drawing for everyone.

Start Drawing

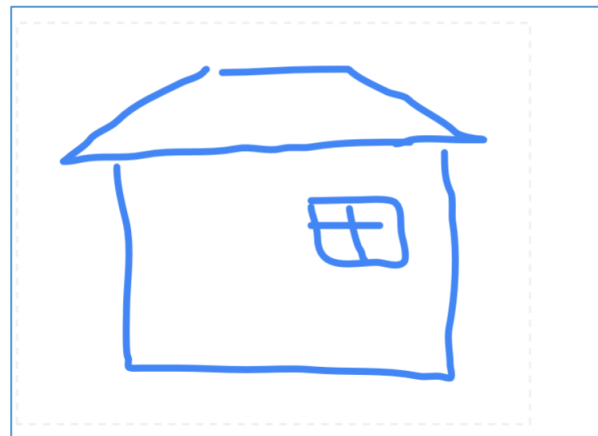
Fast How-To\*

This is an  
A.I.  
Experin

# 人工知能を使ってイラストを描く AutoDraw

③ 描きたいものをざっくり描く

<https://www.autodraw.com/>



④ 上のメニューに候補が出るのでクリック



# 人工知能で、スケッチを増やすサイト

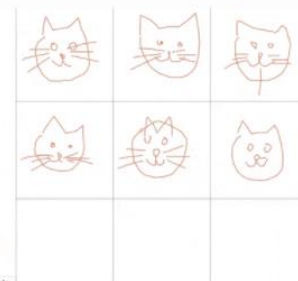
<https://magenta.tensorflow.org/sketch-rnn-demo>

① スクロールして、  
下の方の「Variational Auto-Encoder」を探す

② 「Variational AutoEncoder Demo」をクリック

Variational Auto-Encoder

sketch-rnn cat auto-encoder.



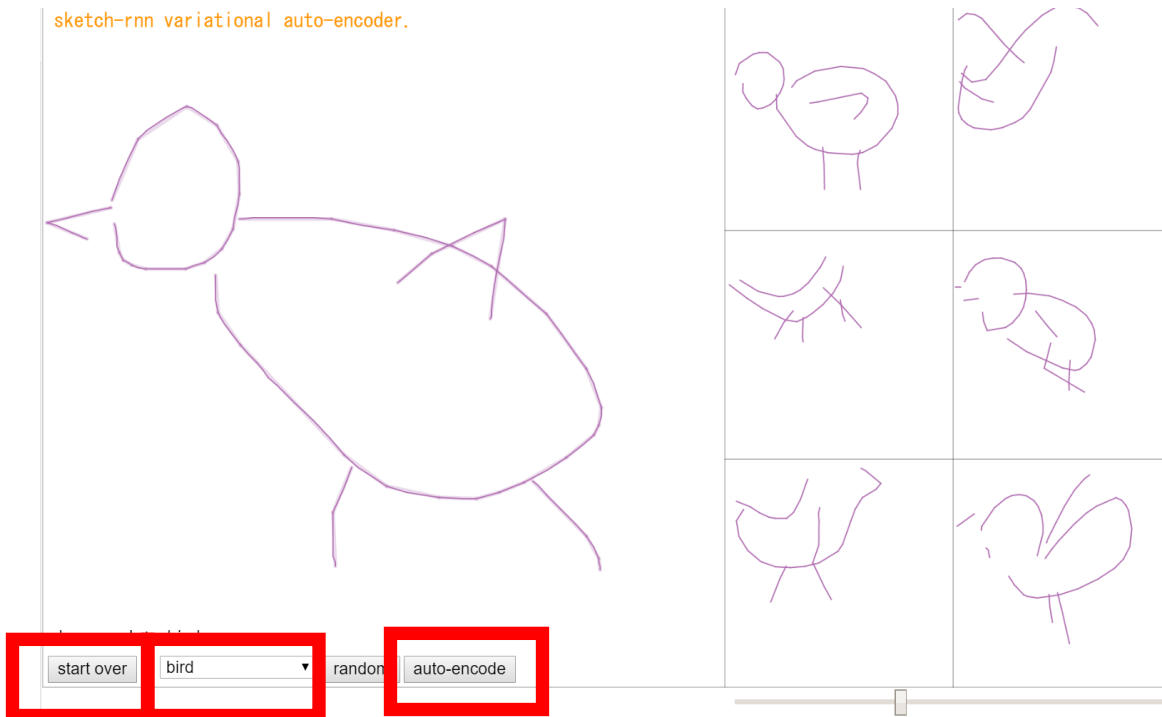
clear drawing cat

Variational Autoencoder Demo

The model can also mimic your drawings and produce similar doodles. In the [Variational Autoencoder Demo](#), you are to draw a **complete** drawing of a specified object. After you draw a complete sketch inside the area on the left, hit the *auto-encode* button and the model will start drawing similar sketches inside the smaller boxes on the right. Rather than drawing a perfect duplicate copy of your drawing, the model will try to mimic your drawing instead.



# 人工知能で、スケッチを増やすサイト



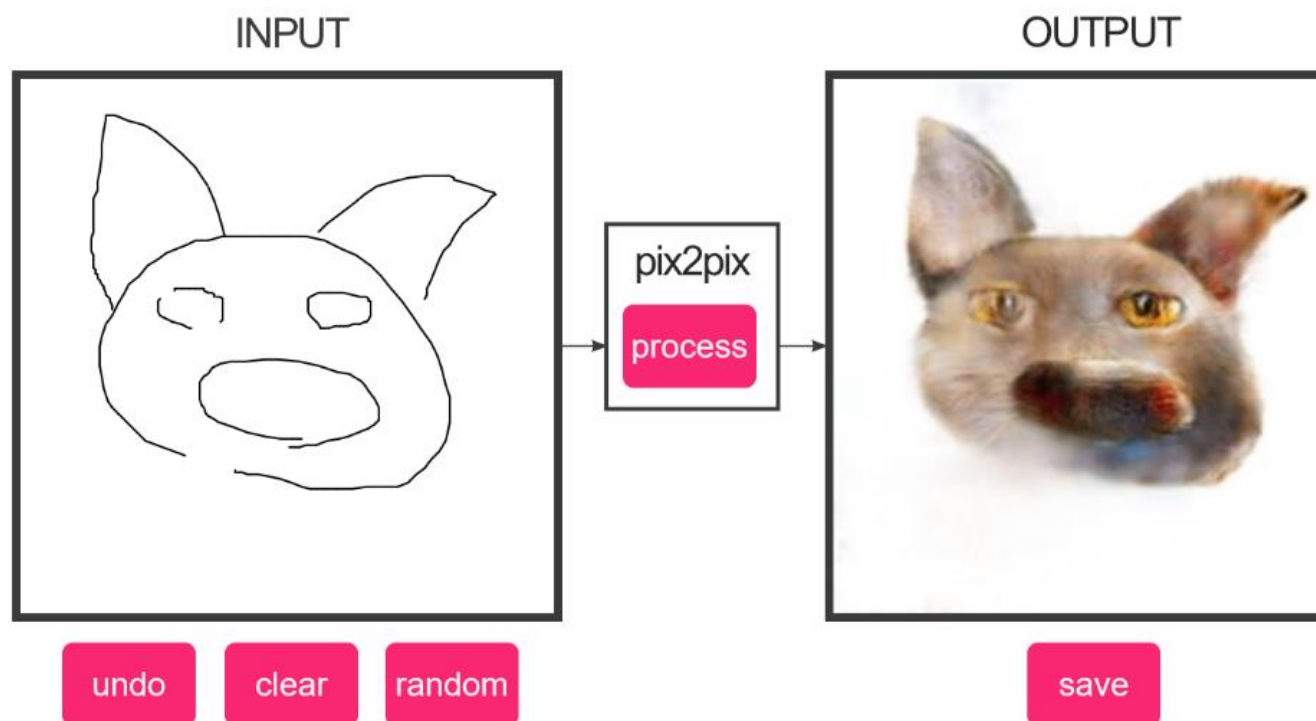
start over  
始めから

種類を  
選べる

auto-encode  
スケッチ生成

# 人工知能で猫を描くウェブサイト

- 「手書き」の絵に合うように、猫の画像を人工知能が描く



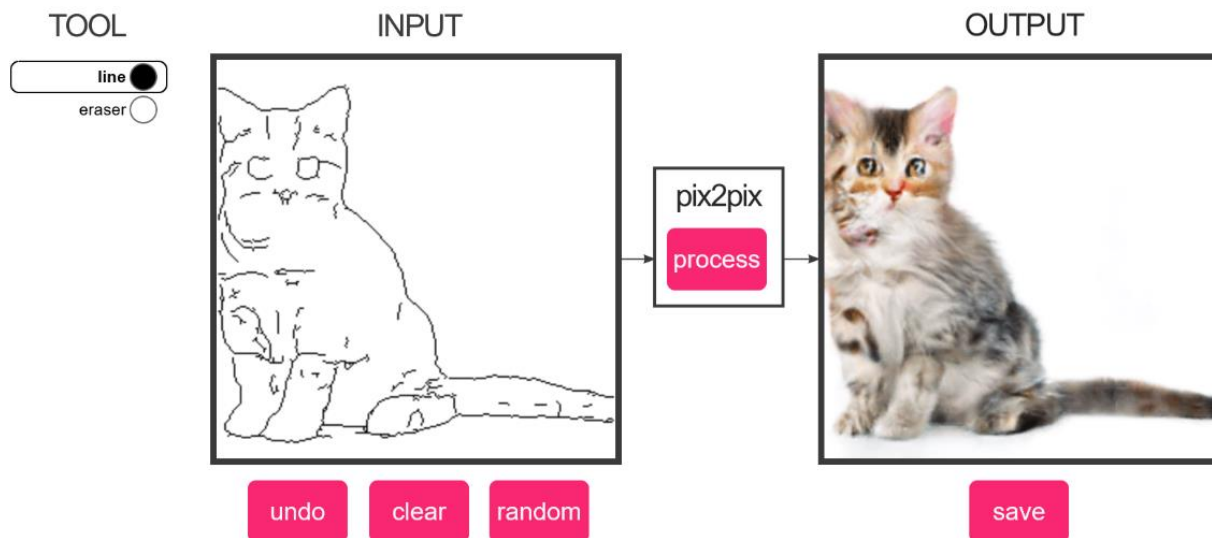
# 人工知能で猫を描くウェブサイト

① ウェブブラウザで次の URL を開く

<http://affinelayer.com/pixsrv/index.html>

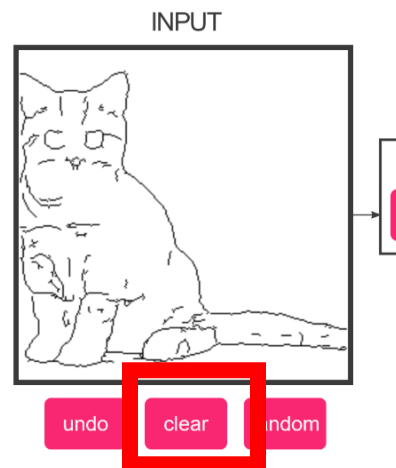
② 「edges2cats」を探す

edges2cats

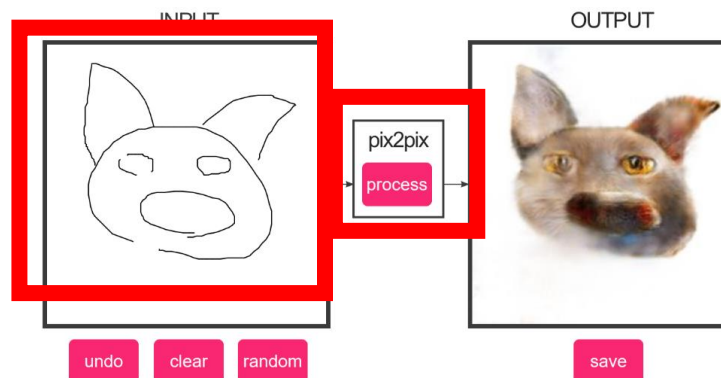


# 人工知能で猫を描くウェブサイト

③ 「**Clear**」 をクリックして消す



④ 猫を手書きして「**process**」をクリック



# 人工知能のデモサイト紹介（パソコン演習）

AIで画像分類を行うデモサイト紹介（clarifai.com）

<https://clarifai.com/demo>

言語 (LANGUAGE) を「日本語 (Japanese)」に切り替えて使用

Demo | Clarifai

https://clarifai.com/demo

clarifai


PRODUCTS ▾ ENTERPRISE ▾ DEVELOPERS ▾ COMPANY ▾ DEMO ▾ PRICING LOG IN

GENERAL FACE NSFW COLOR MORE MODELS ▾

General VIEW DOCS

Japanese (ja)

PREDICTED CONCEPT	PROBABILITY
日暮れ	0.999
暁	0.996
水域	0.995
たそがれ	0.991
夜	0.979
反射	0.979
ポート	0.978
浜	0.972
太陽	0.971



Japanese (ja)  
にしてみる

This website stores cookies on your computer. These cookies are used to collect information about how you interact with our website and allow us to remember you. We use this information in order to improve and customize your browsing experience and for analytics and metrics about our visitors both on this website and other media. To find out more about the cookies we use, see our [Privacy Policy](#).

Got it, thanks.



各自，画像を用意して試してみる

## 6-2 プロダクションシステム

# プロダクションシステム

**ルール**は、あてまめる**条件**と、**条件**が成立したときの状態の**変化**を定めるもの

- **条件**

例 **ichiro** が **ball** を持っていれば

- **変化**

例 **saburo** は **shiro** に **ball** を渡す

**状態データ**は、**作業領域**に記憶され、**ルール**の適用により変化

※ **状態データ**は、**状態**を記述したデータのこと



# プロダクションシステムのルール

**<条件> → <変化>**

**ルールの形**

## 2つのルール

- (1)  $X > Y \rightarrow X$  と  $Y$  の値を入れ替え
- (2)  $X < 0 \rightarrow X$  の値を 0 にする

# プロダクションシステムのルール

## 2つのルール

(1)  $X > Y \rightarrow X$  と  $Y$  の値を入れ替え

(2)  $X < 0 \rightarrow X$  の値を  $0$  にする

初期状態

X	Y
-100	-50

ルール (2) で変化

X	Y
0	-50

ルール (1) で変化

X	Y
-50	0

ルール (2) で変化  
変化を繰り返す

X	Y
0	0

# 確認クイズ

- プロダクションシステム  
ページ 17 の 2 つのルール

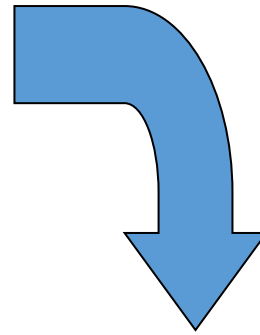
初期状態が  $X = -10, Y = -20$  のとき, どのように変化を繰り返すか, 説明しなさい

	X	Y
初期状態	-10	-20

# コンピュータの入力と出力

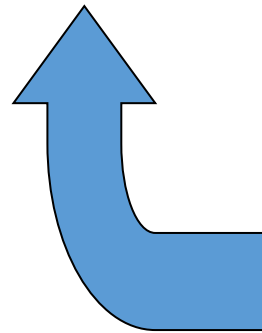
人間や  
他のコンピュータ

入力



コンピュータ

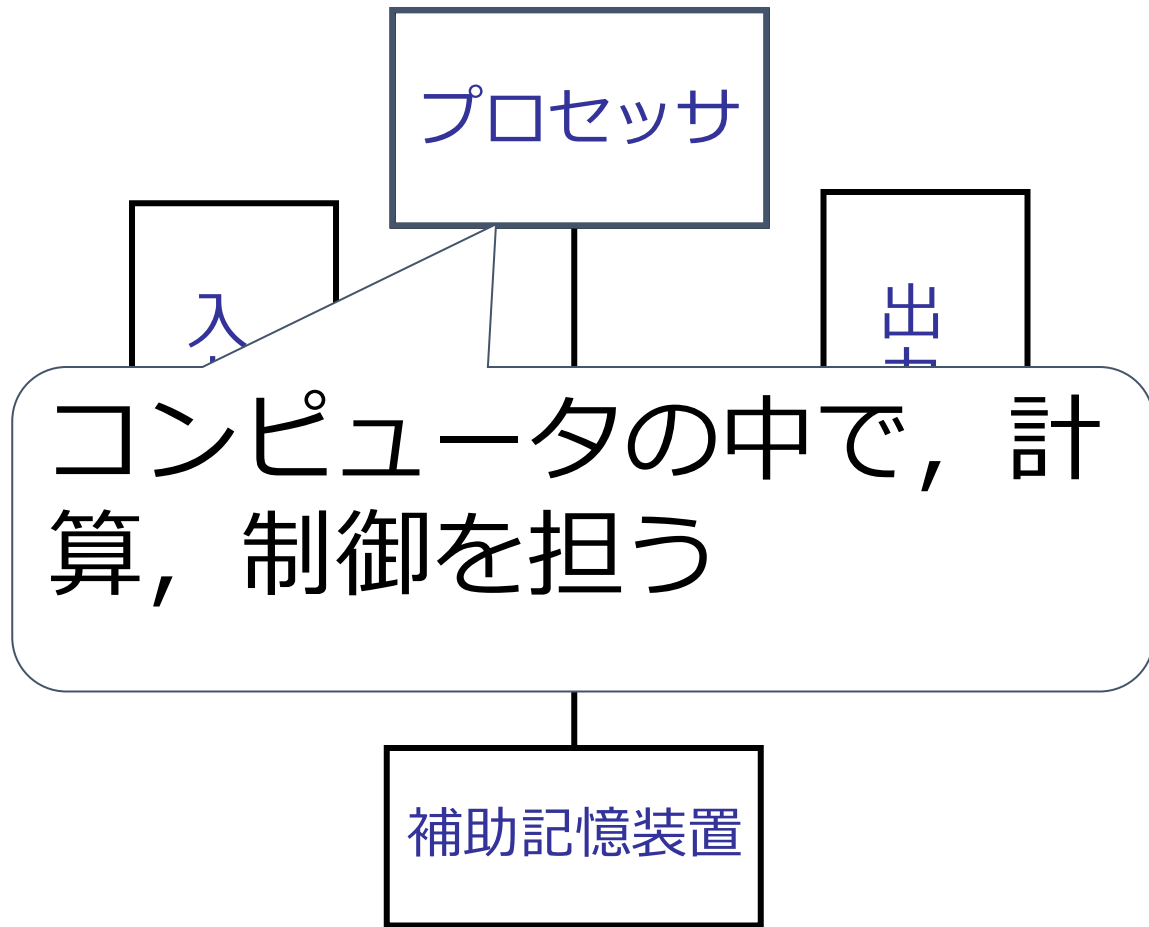
出力



# ハードウェアとソフトウェア

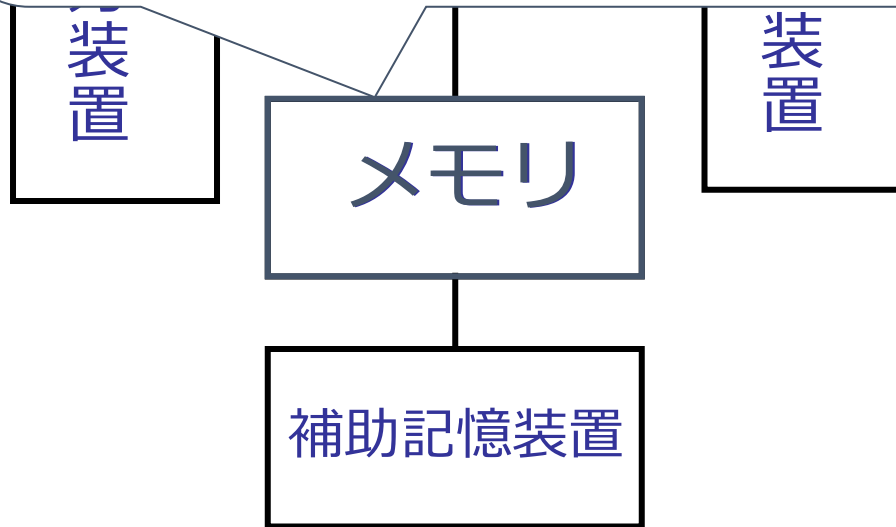
**ソフトウェア**      アプリケーション  
オペレーティングシステム

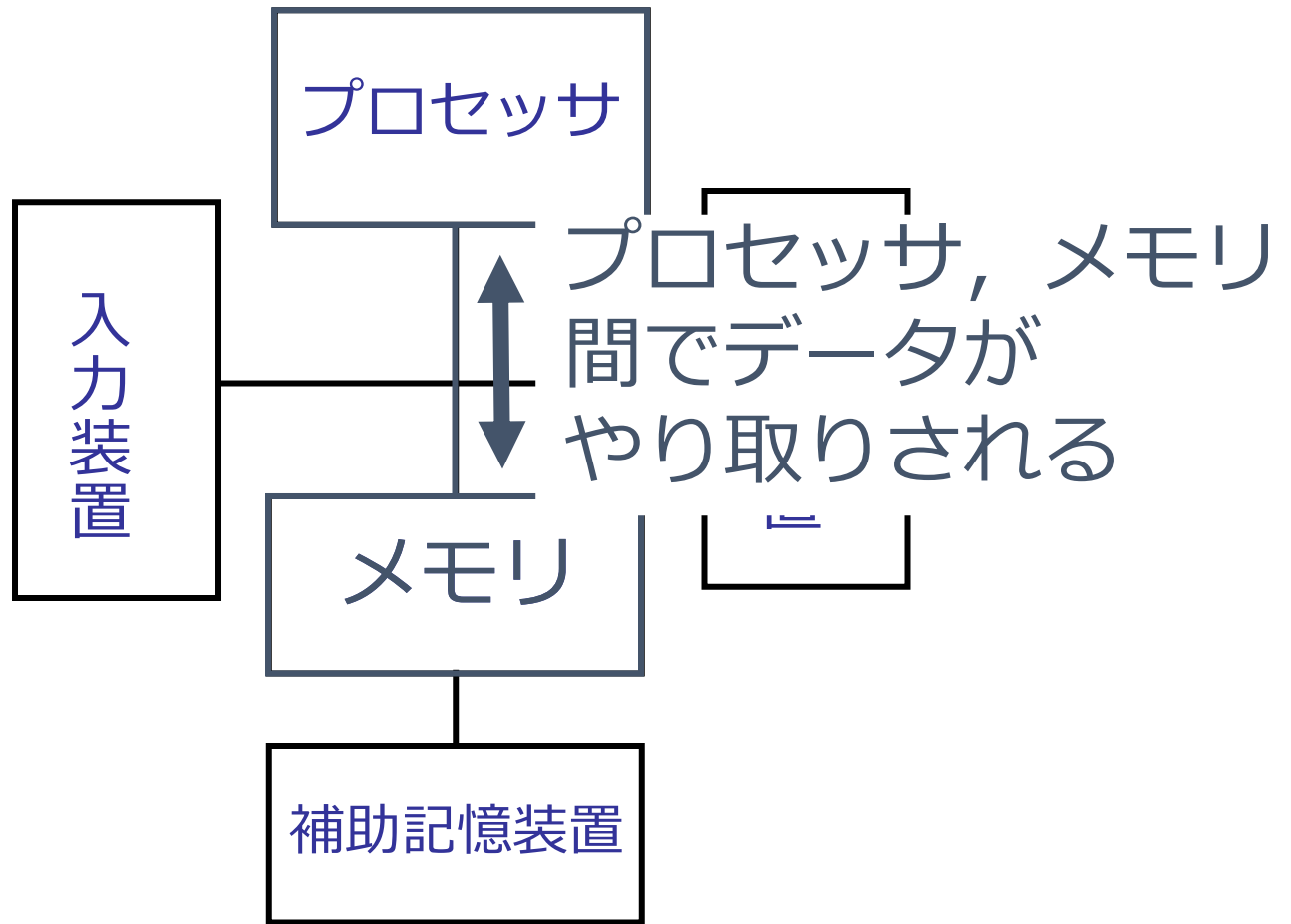
**ハードウェア**      プロセッサ, メモリ  
その他, 入力装置, 出力装置など



データの記憶を行うチップ

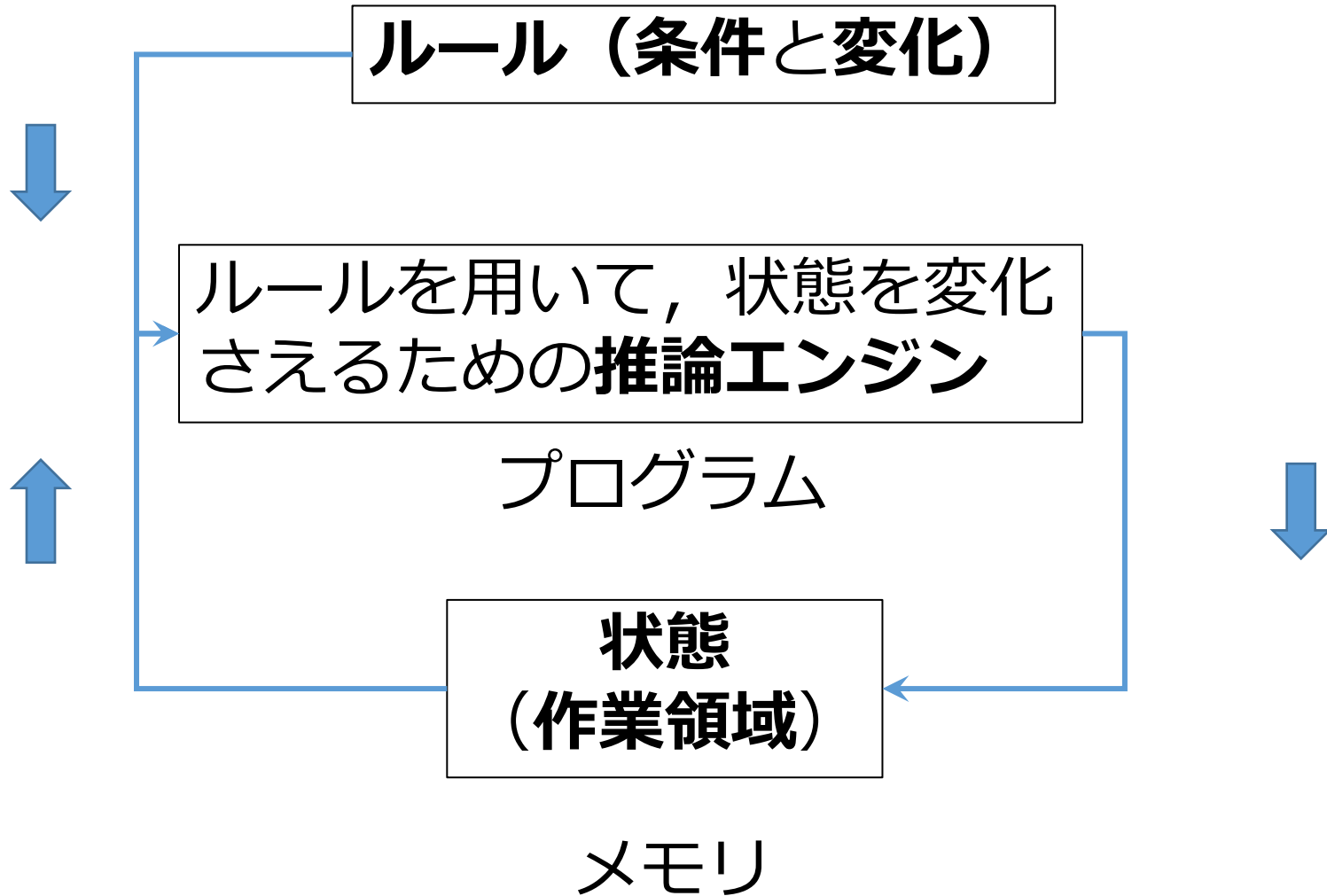
データの書き込み, データの読み出し機能がある







# プロダクションシステムの仕組み



## 作業領域

- **状態データ**を格納する
- **推論エンジン**により，書き換えが行われる

# ルールの形

**<条件> → <変化>**

**ルールの形**

コンピュータのプログラムでは、次のように書く

```
if <条件>:  
    <変化>
```

※ 使用するプログラミング言語により、  
書き方が変わる

# 推論エンジン

- **ルール**を用いて，作業領域内の**状態データ**を書き換える
- 最終的に，結論を得る

# 推論エンジンの仕組み

(1) 作業領域内の状態データを調べる。

状態データがあてはまるような条件を探す

(2) (1) で探した条件について、そこで定められた

変化の通りに、状態データを書き換える

= 推論

(3) 以上を繰り返し、状態データがあてはまるような条件が見つからなくなったら、終了。

結論を得る

## 「変化」の例

- 体毛 = ある → 種類 = 哺乳類

種類

哺乳類

- 種類 = 哺乳類 and 肉食 = する  
→ 種類 = 肉食動物

種類

動物

# 初期状態からの変化の例

- 体毛 = ある → 種類 = 哺乳類
- 種類 = 哺乳類 and 肉食 = する  
→ 種類 = 肉食動物

初期状態	体毛	肉食
	ある	する

変化	体毛	肉食	種類
	ある	する	哺乳類

変化	体毛	肉食	種類
	ある	する	肉食動物

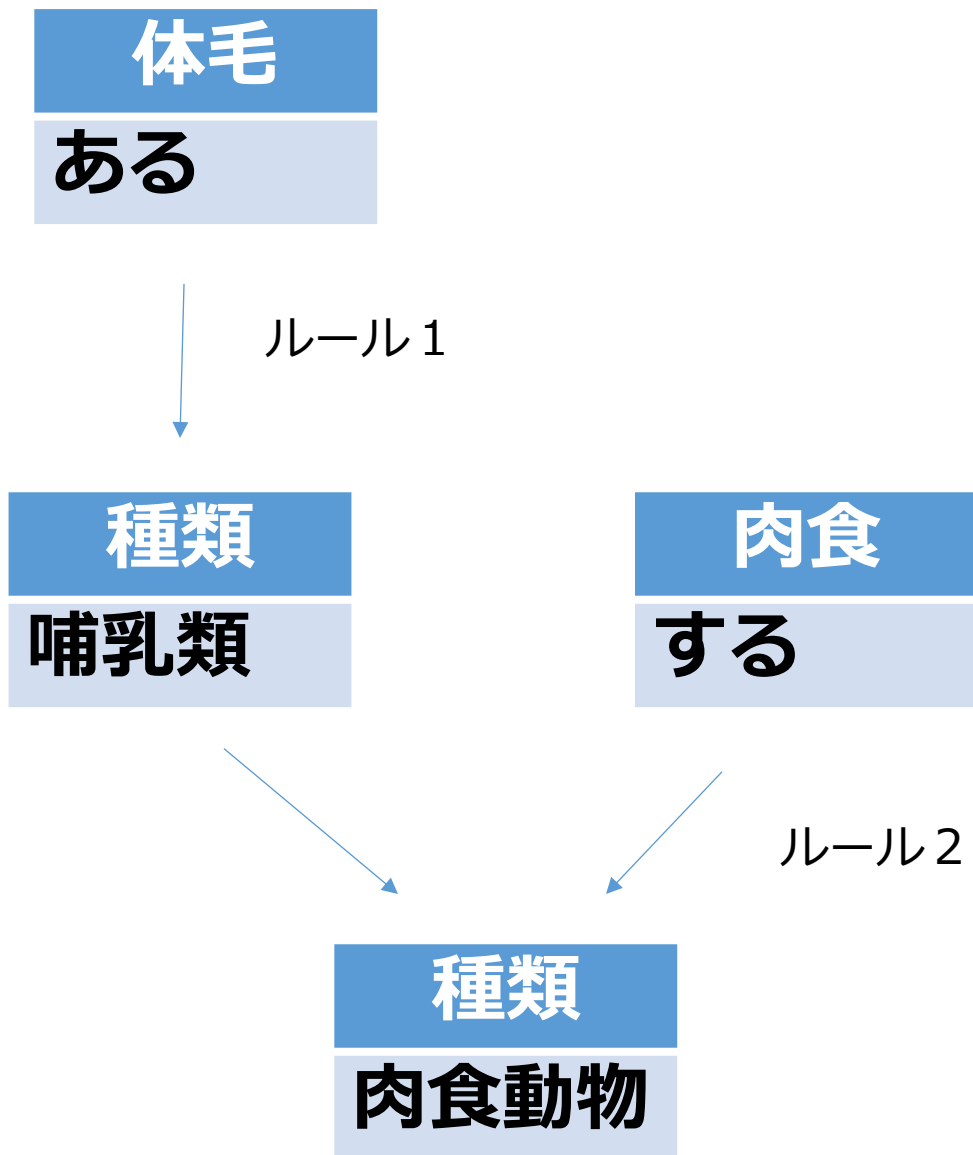
変化を繰り返し，結論を得る

# 推論ネット

- どのように推論が行われたかを示すネットワーク



# 推論ネットの例



# 確認クイズ

- 次の2つのルールがある

ルール1 : 明るさ = 明るい → 昼か夜か = 昼

ルール2 : 昼か夜か = 昼 and 暇か = はい

→ 行動 = 外出

- 次の初期状態について, 推論ネットを書きなさい

明るさ	暇か
明るい	はい

## まとめ

もし～ならば（条件）  
・・・せよ（変化）

- **ルール**に基づく**推論**をマシンに行わせ，問題解決する
- **ルール**を利用して問題解決を行うシステム
  - ⇒ **プロダクションシステム**
  - **ルール**：条件と変化
  - **作業領域**：状態データが置かれる。推論エンジンによる書き換え

## 6-3 プロダクションシステムでの 知識表現

# 知識表現

- ・ プロダクションシステム  
「ルール」による知識表現

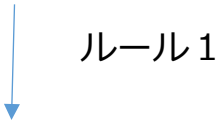
※ 人工知能システムには、多数の知識表現のバリエーションがある

- ・ 論理モデル
- ・ 手続きモデル
- ・ 意味ネットワークモデル
- ・ フレームモデル

# プロダクションシステムの知識表現

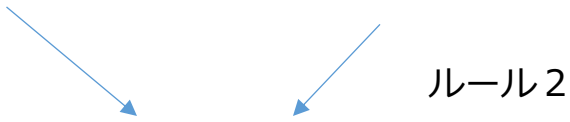
- **推論**の仕組みにより, **知識**を扱うマシンを実現可能
- 種々の利点
  1. 知識を**ルール**を形式で表現. 自然で理解しやすい.
  2. **条件**と**変化**の両方を記述できる.
  3. **ルール**の変更, 追加, 削除が, 他のシステム (前ページ) より簡単とされている.
- 種々の欠点
  1. すべての**推論**を, **ルール**の当てはめの繰り返しで行う.  
    **ルール**数が莫大なとき, 処理が遅い心配
  2. **結論**しか得られない. 次にどう行動すべきかの**探索**ではない.  
    ※ **探索**については, 以前の授業で学習済み

**明るさ**  
明るい



**昼か夜か**  
昼

**暇か**  
はい



**行動**  
外出

X	Y
-10	-20
X	Y
0	-20
X	Y
-20	0
X	Y
0	0

正解例の1

X	Y
-10	-20
X	Y
-20	-10
X	Y
0	-10
X	Y
-10	0
X	Y
0	0

正解例の2