

# CNNによる教師あり学習を用いた Samurai Coding用プログラムの試作

日本大学 文理学部 情報科学科 谷研究室  
井上 尚大

# 目次

- ▶ 1 : はじめに
- ▶ 2 : 機械学習におけるゲームAI
- ▶ 3 : 演習目的・内容
- ▶ 4 : 演習方法
- ▶ 5 : 演習結果・考察
- ▶ 6 : まとめ

# 目次

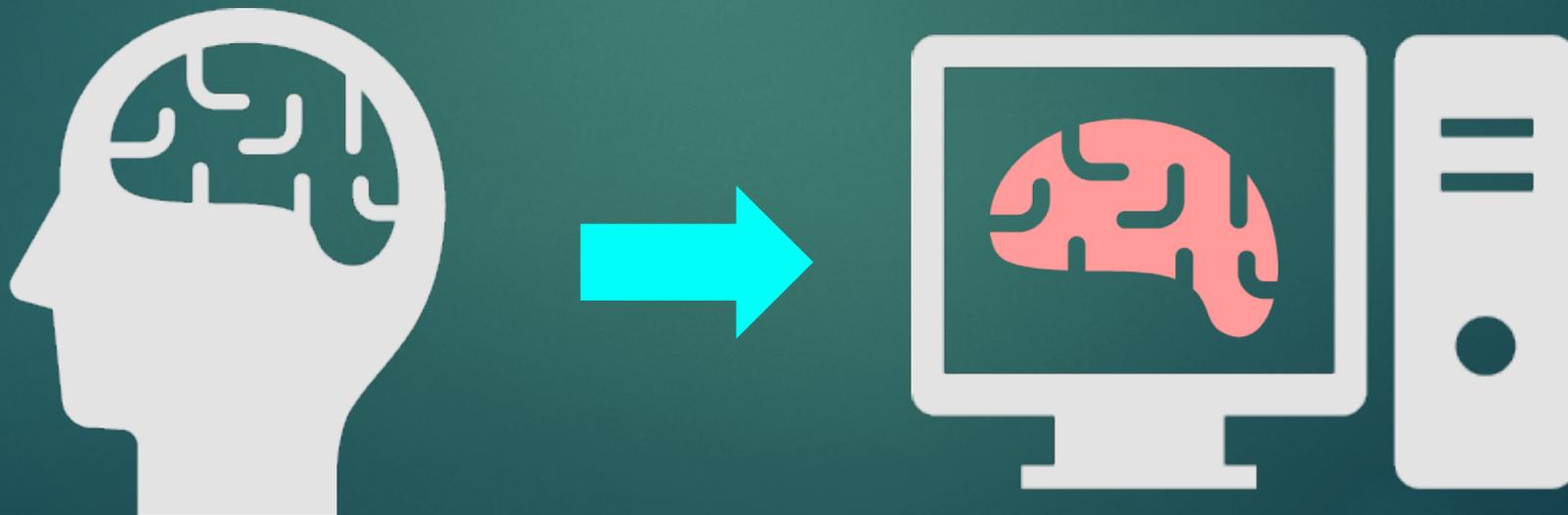
- ▶ 1 : はじめに
- ▶ 2 : 機械学習におけるゲームAI
- ▶ 3 : 演習目的・内容
- ▶ 4 : 演習方法
- ▶ 5 : 演習結果
- ▶ 6 : まとめ

# 1 : はじめに

- ▶ 1 : 人工知能
- ▶ 2 : 機械学習
- ▶ 3 : 深層学習

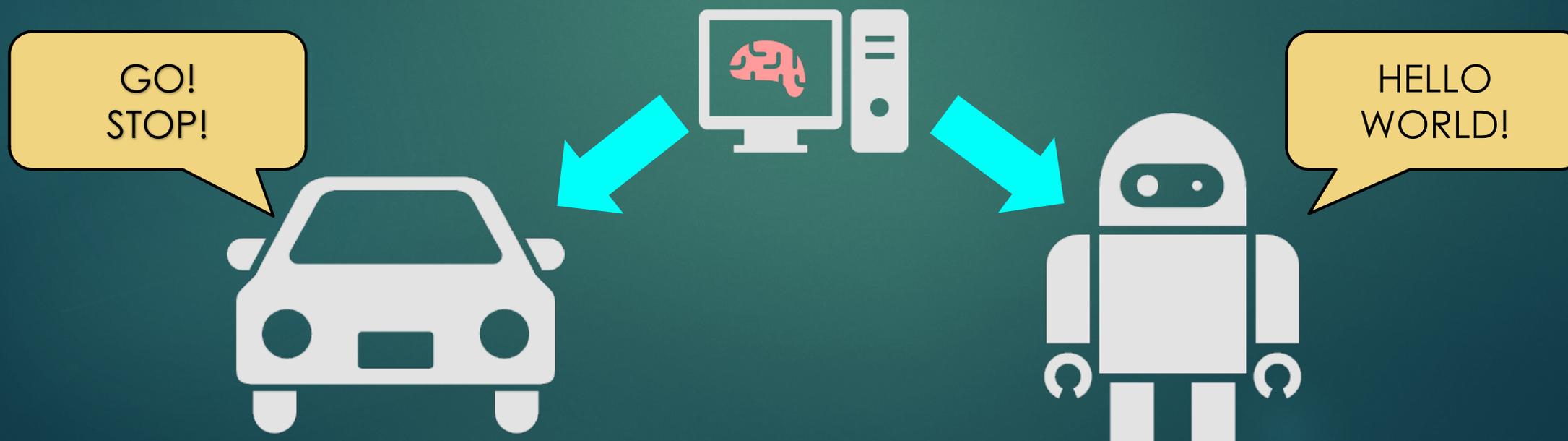
# 1 : はじめに

- ▶ 1 : 人工知能 (Artificial Intelligence)
- ▶ 人工的にコンピュータ上で人間と同様の知能を実現させる試み (wikipedia)



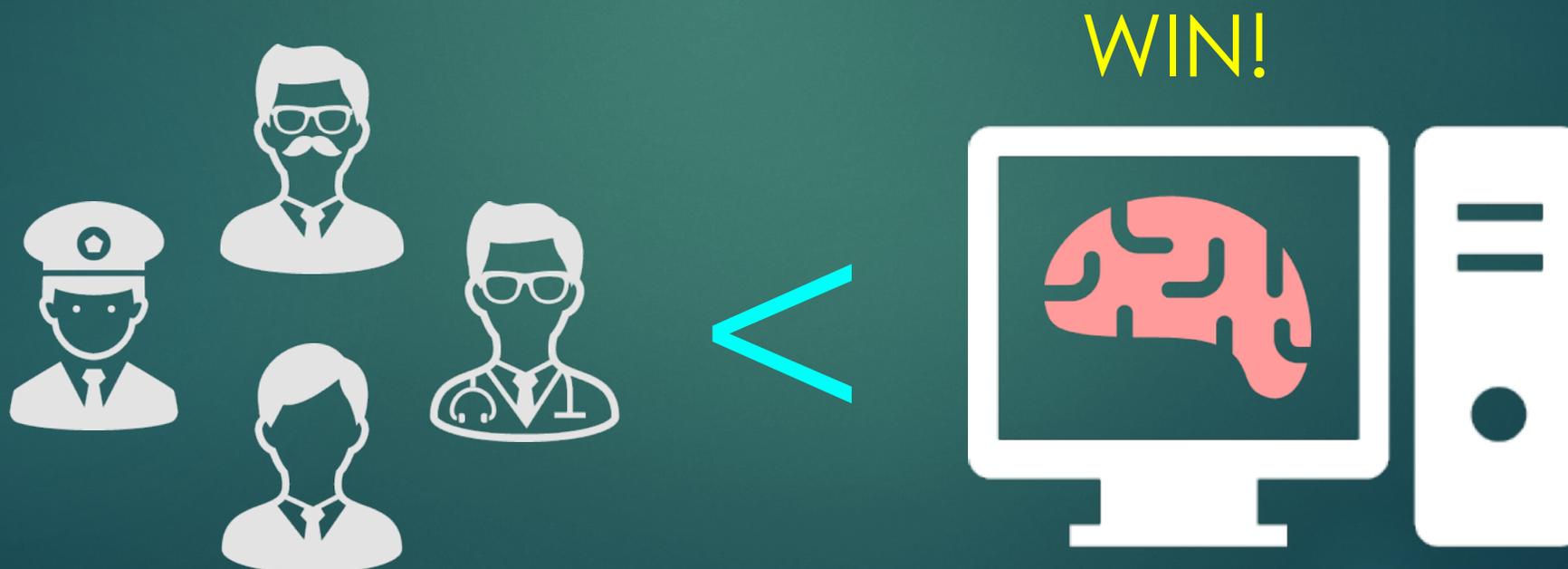
# 1 : はじめに

- ▶ 1 : 人工知能 (Artificial Intelligence)
- ▶ その活用範囲は医療、福祉、教育と様々  
身近な例 : 自動運転、チャットボット



# 1 : はじめに

- ▶ 1 : 人工知能 (Artificial Intelligence)
- ▶ 人工知能の発達により人々の暮らしは便利になる一方で、「2045年問題」が懸念される



# 1 : はじめに

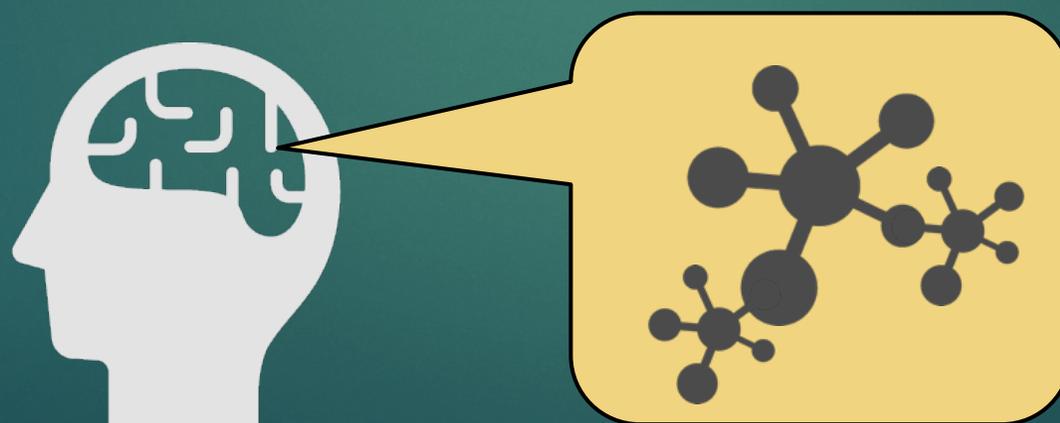
- ▶ 2 : 機械学習 (Machine Learning)
- ▶ 明示的にプログラムせず、学習する能力をコンピュータに与える  
研究分野 (Arthur Lee Samuel 1959)
- ▶ 人間が自然に行う学習能力と同様の機能をコンピュータで用い、  
実現する技術、手法 (Wikipedia)

# 1 : はじめに

- ▶ 3 : 深層学習 (Deep Learning)
- ▶ 学習精度の向上を目的とした多層構造ニューラルネットワークを用いる機械学習技術  
(「AIの未解決問題とDeep Learning」 東京大学 松尾豊 著)
- ▶ 画像認識、自然言語処理、音声認識、ゲームなど様々な分野で高い精度を誇る

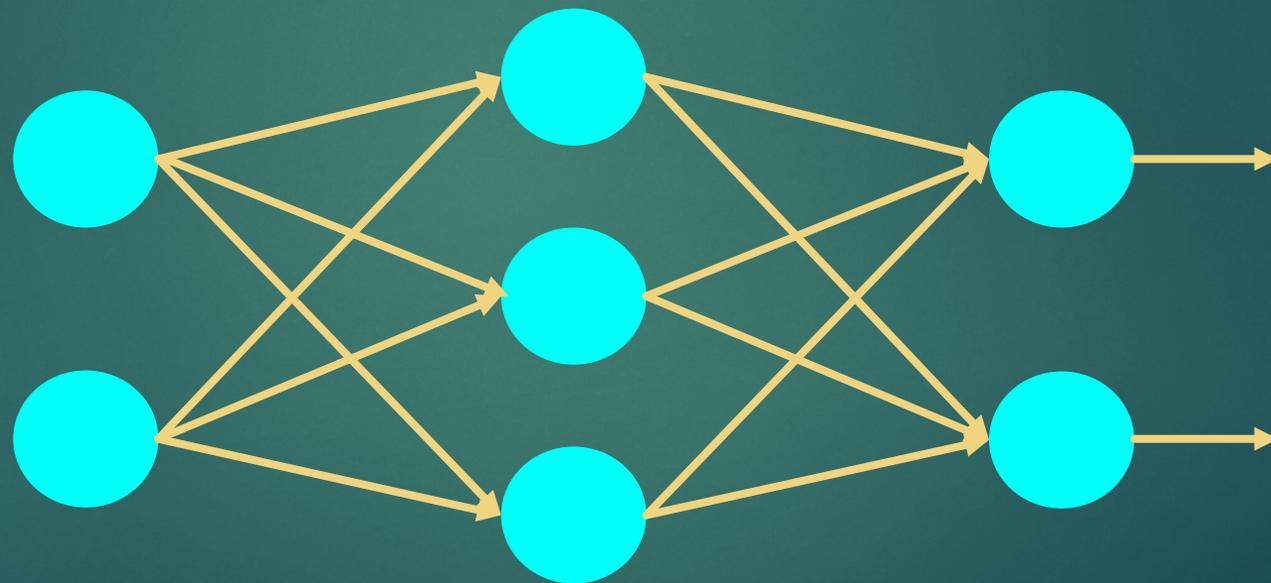
# 1 : はじめに

- ▶ ニューラルネットワーク (Neural Network)
- ▶ 脳内のニューロン (神経細胞) の仕組みをコンピュータ内に実現しようとしたもの  
(静岡理工科大学情報学部菅沼義昇研究室HP)



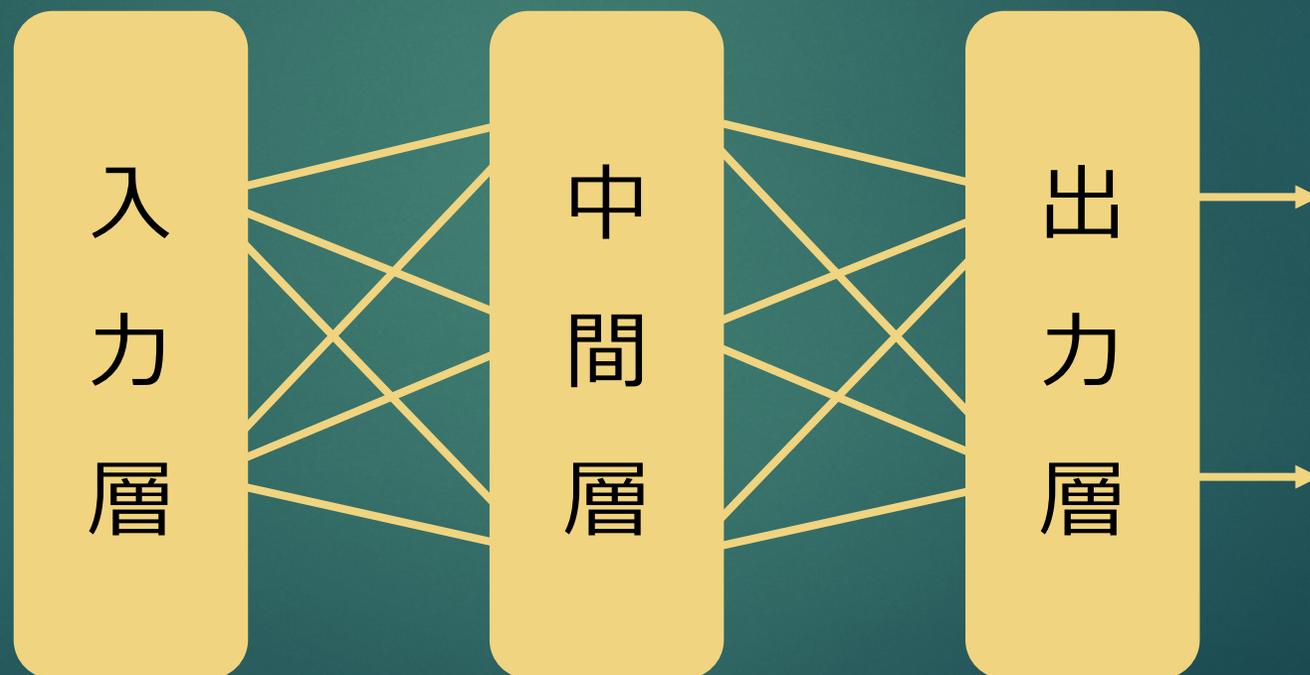
# 1 : はじめに

## ▶ ニューラルネットワーク (Neural Network)



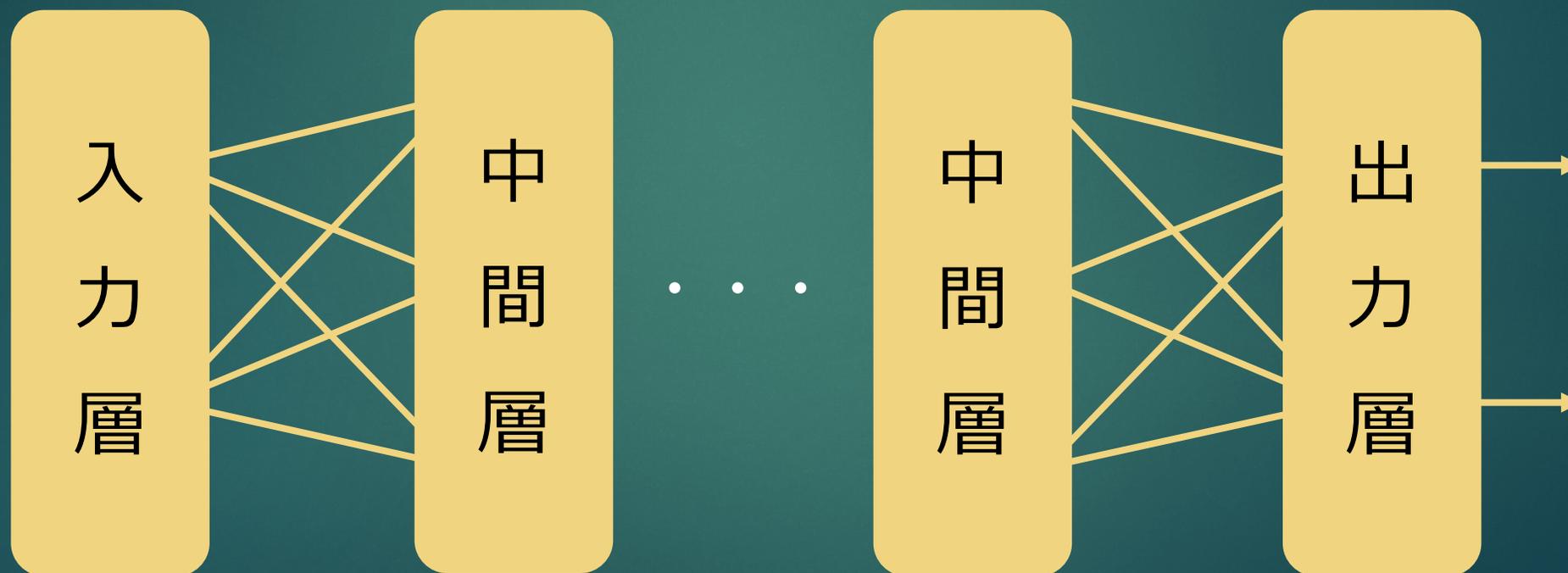
# 1 : はじめに

## ▶ ニューラルネットワーク (Neural Network)

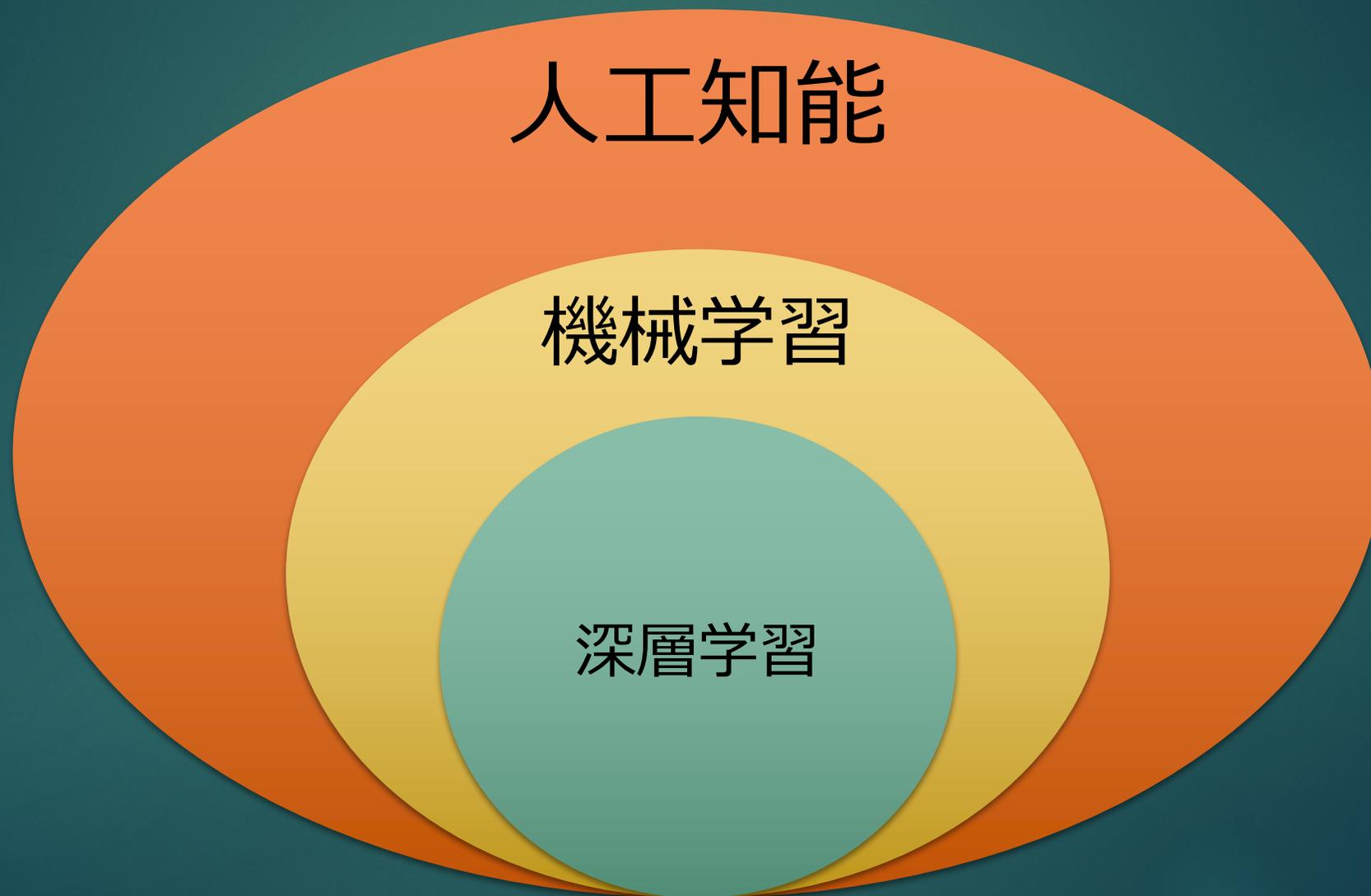


# 1 : はじめに

## ▶ 多層ニューラルネットワーク



# 1 : はじめに



# 目次

- ▶ 1 : はじめに
- ▶ 2 : 機械学習におけるゲームAI
- ▶ 3 : 演習内容・目的
- ▶ 4 : 演習方法
- ▶ 5 : 演習結果・考察
- ▶ 6 : まとめ

## 2 : 機械学習におけるゲームAI

### ▶ 概要

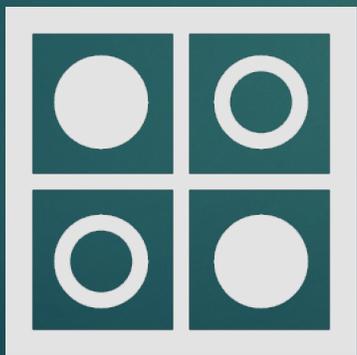
- ▶ ゲームの分野においても人工知能は用いられている (ゲームAI)



## 2 : 機械学習におけるゲームAI

### ▶ 歴史

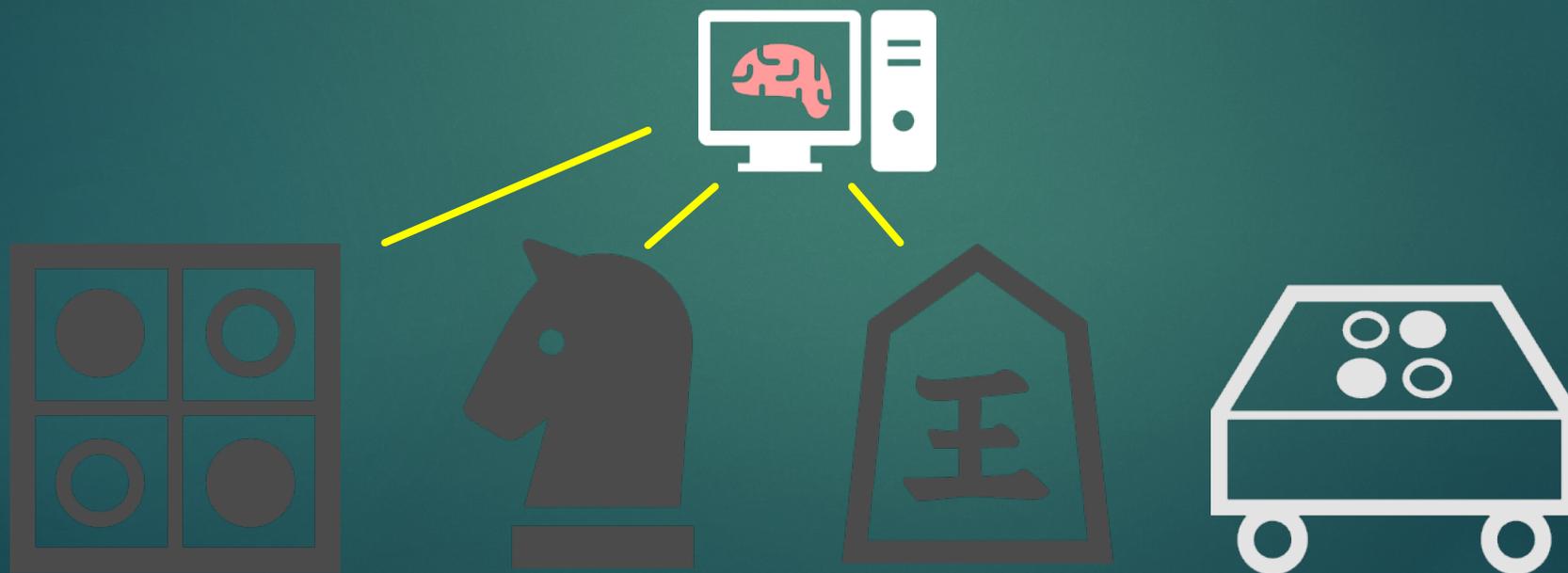
- ▶ 1950年代頃から研究が開始
- ▶ 単純なゲームだけでなく複雑なゲームへの研究も進行



## 2 : 機械学習におけるゲームAI

### ▶ 歴史

- ▶ 複雑なゲームに対して機械学習を取り入れることにより、世界のトッププロの選手に勝利



## 2 : 機械学習におけるゲームAI

### ▶ 現状

- ▶ 2016年3月にGoogle（現Alphabet）の子会社DeepMindの作った囲碁AI「AlphaGo」が世界トップのプロ棋士に勝利し話題となる



## 2 : 機械学習におけるゲームAI

### ▶ 現状

- ▶ 現在国内で多くのゲームAIの大会が開催されている



## 2 : 機械学習におけるゲームAI

### ▶ 現状

- ▶ 中でも機械学習を取り入れたプログラムが大会の上位を占める



## 2 : 機械学習におけるゲームAI

### ▶ 動機

- ▶ ゲームAIの大会に「AlphaGo」の仕組みを取り入れ優勝可能か？



# 目次

- ▶ 1 : はじめに
- ▶ 2 : 機械学習におけるゲームAI
- ▶ 3 : 演習目的・内容
- ▶ 4 : 演習方法
- ▶ 5 : 演習結果・考察
- ▶ 6 : まとめ

# 3 : 演習目的・内容

## ▶ 目的

- ▶ 囲碁以外のゲームにおいても「AlphaGo」の仕組みが応用可能であるが、武行

- ▶ 情報処理学会主催の「Samurai Coding」



# 3 : 演習目的・内容

## ▶ 内容

- ▶ 「AlphaGo」の仕組みの一部を模した大会用プログラムの作成

## ▶ 評価方法

- ▶ 昨年の大会上位入賞者のプログラムとの対戦

# 3 : 演習目的・内容

- ▶ AlphaGoの仕組み
- ▶ 1 : 教師あり学習フェーズ
- ▶ 2 : 強化学習フェーズ
- ▶ 3 : 評価関数作成フェーズ
- ▶ 4 : モンテカルロ木探索フェーズ

# 3 : 演習目的・内容

- ▶ AlphaGoの仕組み
- ▶ 1 : 教師あり学習フェーズ
- ▶ 2 : 強化学習フェーズ
- ▶ 3 : 評価関数作成フェーズ
- ▶ 4 : モンテカルロ木探索フェーズ

打ち手を予測

勝てる手を予測

盤面を評価

数手先を読む

# 3 : 演習目的・内容

## ▶ AlphaGoの仕組み

### ▶ 1 : 教師あり学習フェーズ

### 2 : 強化学習フェーズ

### 3 : 評価関数作成フェーズ

### 4 : モンテカルロ木探索フェーズ

打ち手を予測

勝てる手を予測

盤面を評価

数手先を読む

# 3 : 演習目的・内容

## ▶ AlphaGoの仕組み

### ▶ 1 : 教師あり学習フェーズ

#### ▶ 1-1 : Rollout policy



高速・低精度な予測

#### ▶ 1-2 : Supervised learning (SL) policy network



高精度な予測

打ち手を予測

# 3 : 演習目的・内容

## ▶ AlphaGoの仕組み

### ▶ 1 : 教師あり学習フェーズ

#### ▶ 1-1 : Rollout policy



高速・低精度な予測

#### ▶ 1-2 : Supervised learning (SL) policy network



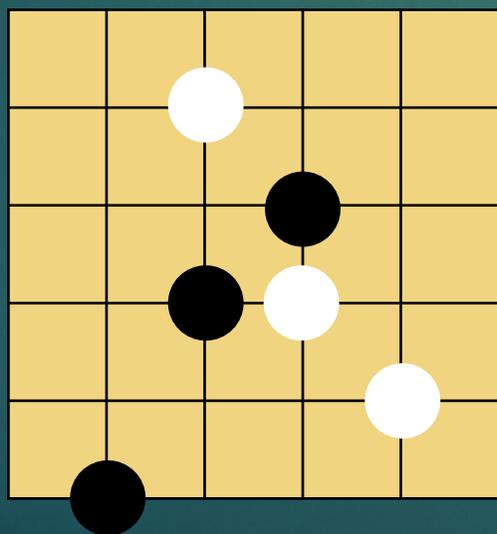
高精度な予測

打ち手を予測

# 3 : 演習目的・内容

## ▶ SL policy network

- ▶ 入力 : 19 × 19の48bit画像に見立てたフラグの集まり
- ▶ フラグ : 自分の石、相手の石、前手で打たれた石など48種類



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# 3 : 演習目的・内容

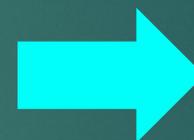
▶ SL policy network

▶ 出力：次の一手の選択確率

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$



SL policy  
network



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0.2 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# 3 : 演習目的・内容

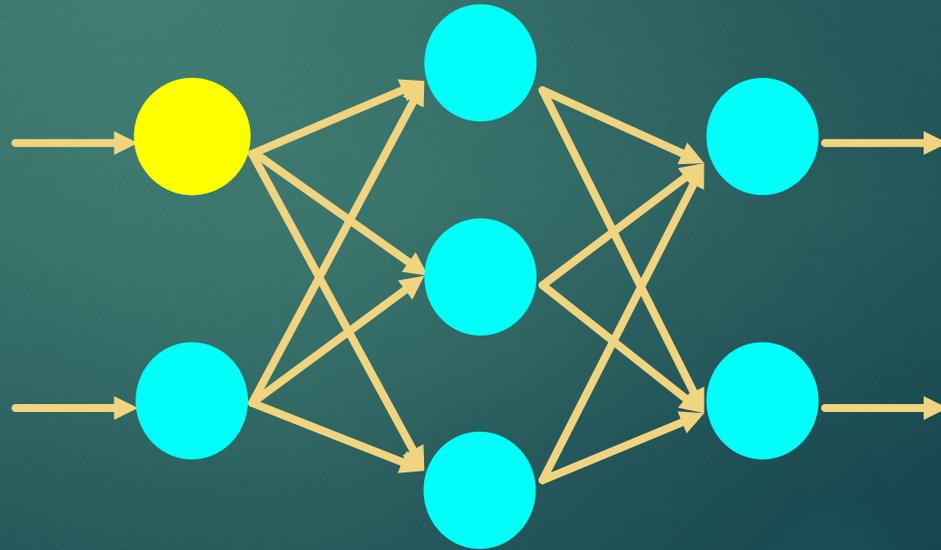
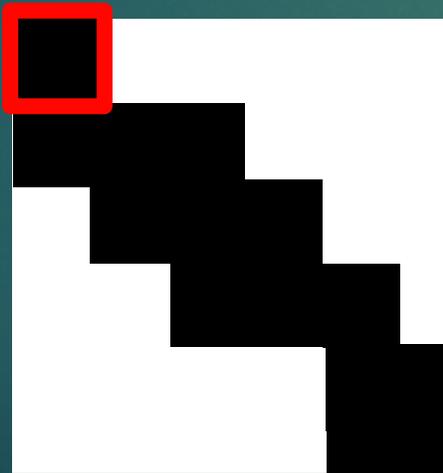
- ▶ 教師あり学習フェーズ
- ▶ 教師データ : KGSの囲碁サーバーにある16万局、2億4千万の盤面
- ▶ 学習内容 : ある盤面での打ち手
- ▶ **学習手法 : CNN**    本演習に用いる手法

# 3 : 演習目的・内容

- ▶ CNN (Convolutional Neural Network)
- ▶ 畳み込みニューラルネットワーク
- ▶ 深層学習の手法の一つ
- ▶ 画像認識で特に優れた実績を収めている

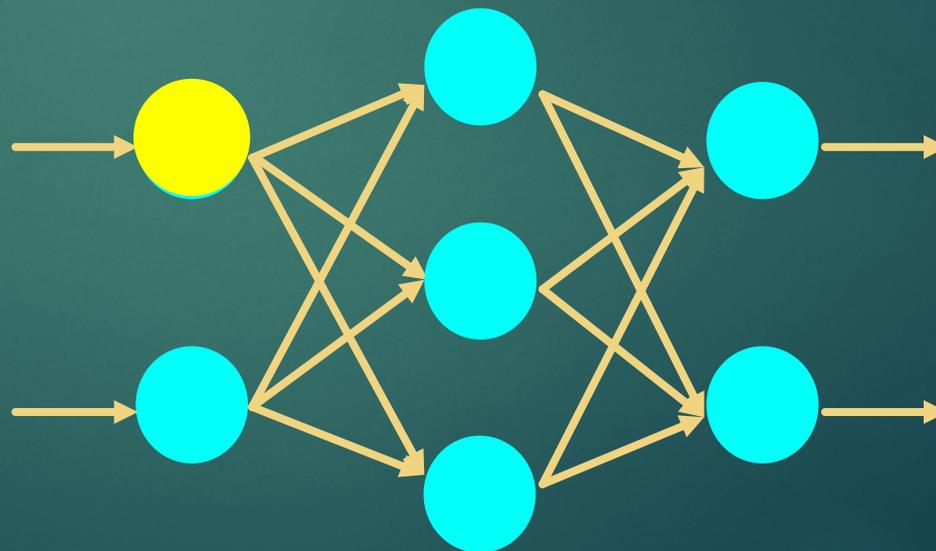
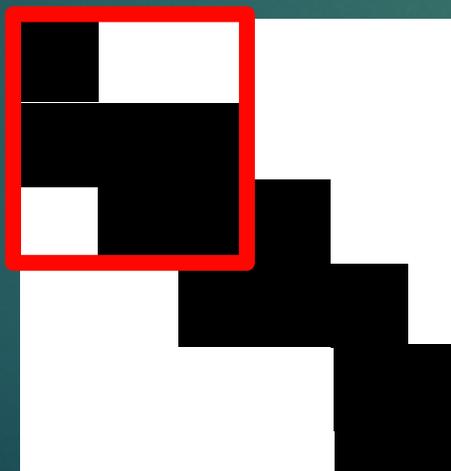
# 3 : 演習目的・内容

- ▶ CNN (Convolutional Neural Network)
- ▶ 通常のニューラルネットワーク  
→ 1ピクセルを一つの入力とする



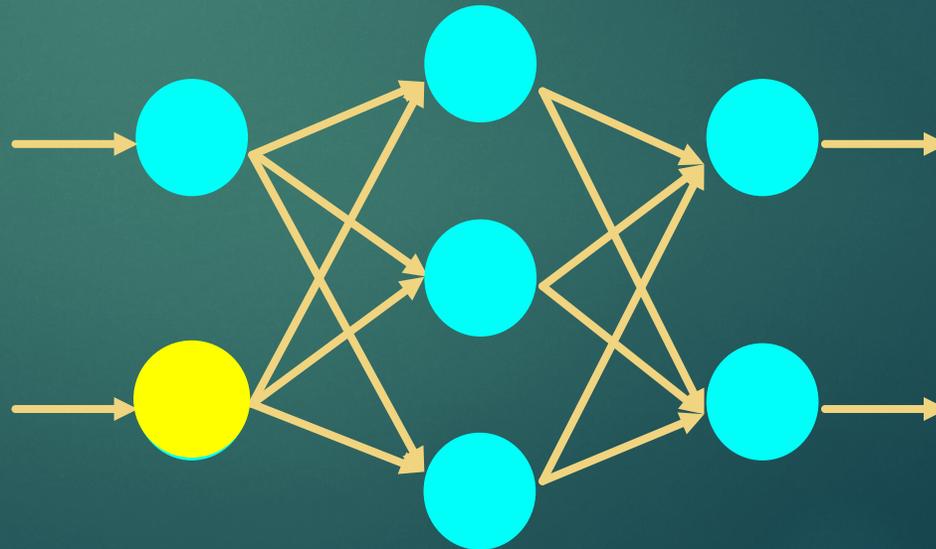
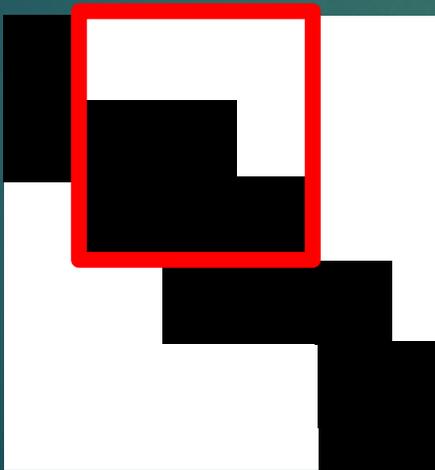
# 3 : 演習目的・内容

- ▶ CNN (Convolutional Neural Network)
- ▶ CNN  
→ 小領域 (フィルタ) を一つの入力とする



# 3 : 演習目的・内容

- ▶ CNN (Convolutional Neural Network)
- ▶ CNN  
→ 多くの特徴量を得ることができるため精度が向上



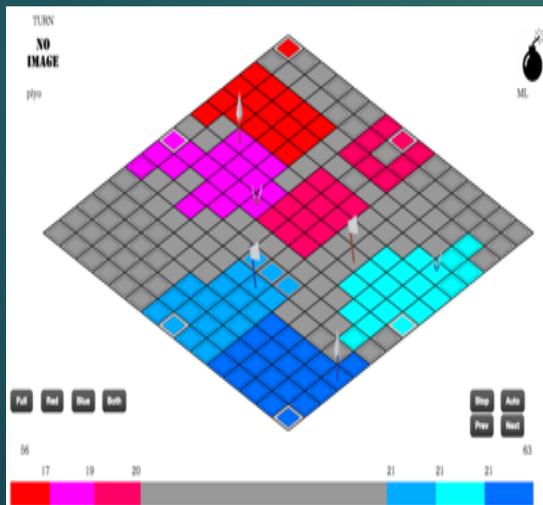
# 目次

- ▶ 1 : はじめに
- ▶ 2 : 機械学習におけるゲームAI
- ▶ 3 : 演習目的・内容
- ▶ 4 : 演習方法
- ▶ 5 : 演習結果・考察
- ▶ 6 : まとめ

# 4：演習方法

## ▶ 演習の要点①

- ▶ 盤面の情報を入力として、行動を決定する仕組みの実現



教師あり学習

?

行動

# 4：演習方法

## ▶ 演習の要点②

- ▶ AlphaGoと違い、png形式の画像による教師あり学習を行う
- ▶ 教師データ：昨年大会上位者同士の対戦データ（70000の盤面）に各盤面で実際にとった行動をラベルとし付与したもの
- ▶ 学習内容：ある盤面をとった行動

# 4 : 演習方法

## ▶ 演習環境

	MacBook	MacBook Pro	Mac mini
OS	Sierra	Sierra	Yosemite
CPU	Intel Core m7	Intel Core i7	Intel Core i5
モデル	Early 2016	Late 2016	Late 2014
クロック	1.3 GHz	2.7 GHz	2.8 GHz
メモリ	8 GB	16 GB	16 GB

# 4 : 演習方法

- ▶ 演習環境
- ▶ 言語 : Python
- ▶ ツール : TensorFlow , OpenCV

# 4 : 演習方法

## ▶ 演習手順

- ▶ 1 : ログデータ収集・整形
- ▶ 2 : CNNを用いた教師あり学習
- ▶ 3 : 大会用プログラムに組み込み対戦

# 4 : 演習方法

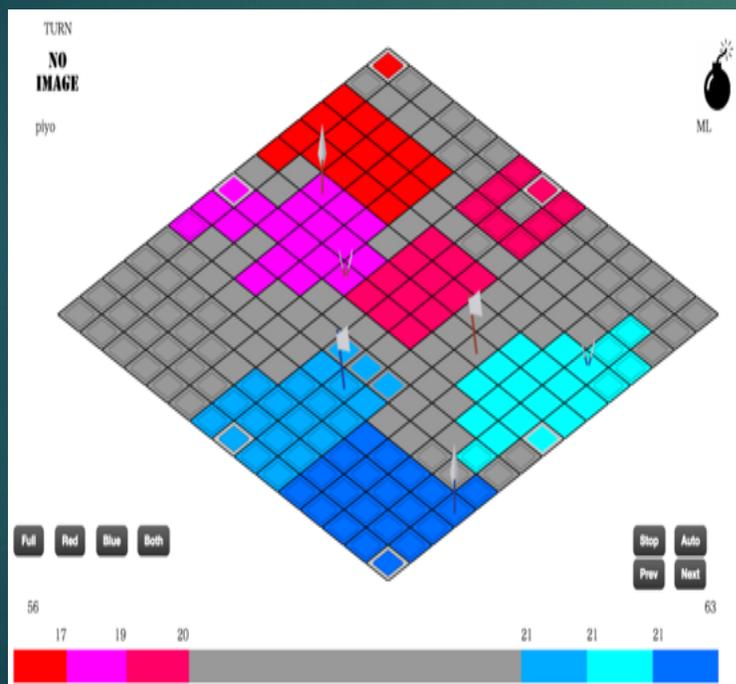
## ▶ 演習手順

- ▶ 1 : ログデータ収集・整形
- ▶ 2 : CNNを用いた教師あり学習
- ▶ 3 : 大会用プログラムに組み込み対戦

# 4 : 演習方法

## ▶ 1 : ログデータ収集・整形

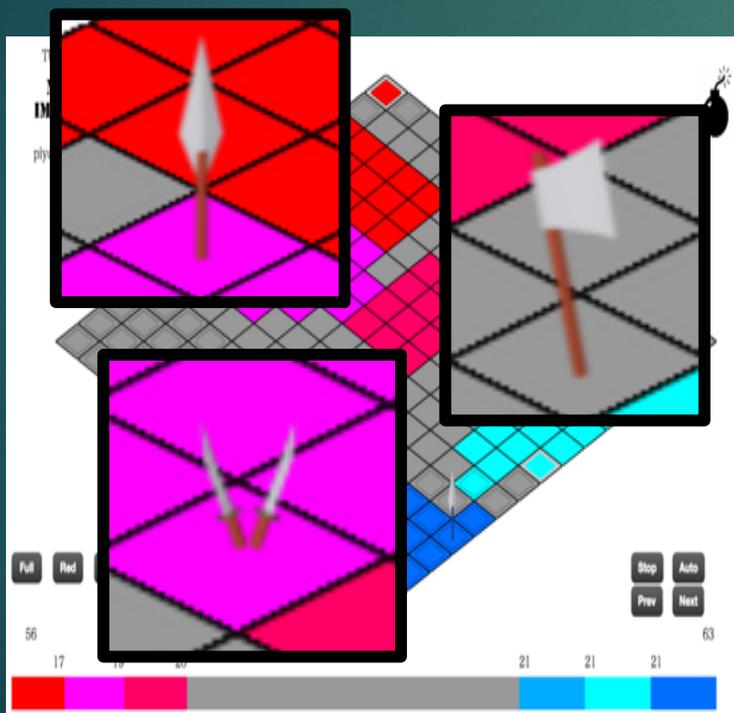
▶ 昨年の大会上位者同士を対戦させ、ログを収集（計10,000試合）



```
# Turn information for samurai of side 0 with weapon 0
# <turn>
84
# <cure period>
0
# Samurai states: <position x> <position y> <hiding>
6 10 0
7 7 0
7 0 0
-1 -1 1
-1 -1 1
7 14 0
# Battle field states
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 2 8 2 8 8 8 9 9 9
9 9 9 9 0 8 2 2 2 8 8 9 9 9 9
9 9 9 9 9 8 2 2 2 4 9 9 9 9 9
9 9 9 9 9 8 2 2 2 2 9 9 9 9 9
9 9 9 9 1 8 2 2 2 2 2 9 9 9 9
9 9 9 1 1 1 2 2 2 2 2 4 9 9 9
9 9 1 1 1 1 2 2 2 3 3 3 3 9 9
9 9 9 1 1 1 1 0 0 0 0 4 9 9 9
9 9 0 1 1 1 1 0 0 0 0 9 9 9 9
9 0 0 8 1 1 1 1 1 3 3 3 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9 9
9 9 9 8 8 8 0 1 1 1 9 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9 9
```

# 4：演習方法

- ▶ 1：ログデータ収集・整形
- ▶ 昨年の大会上位者同士を対戦させ、ログを収集（計10,000試合）

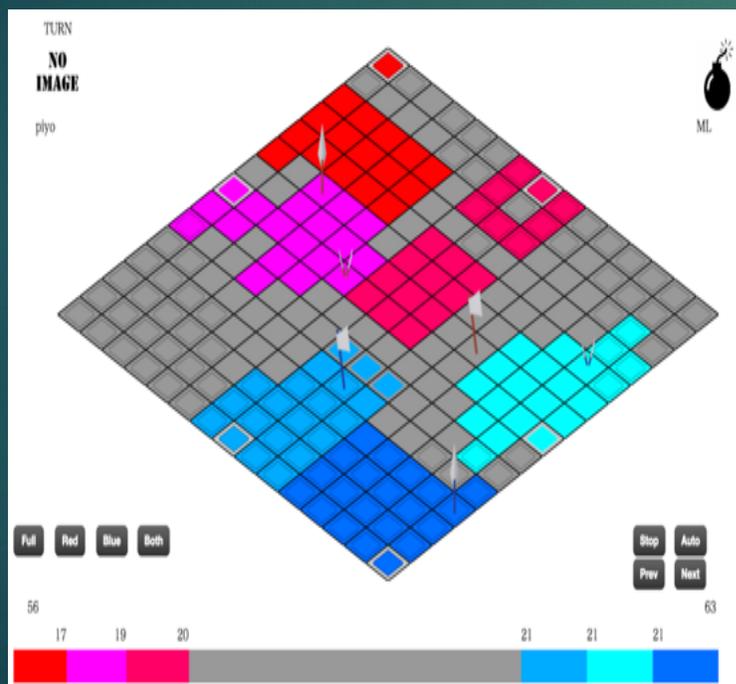


味方・敵の位置

# 4 : 演習方法

## ▶ 1 : ログデータ収集・整形

- ▶ 昨年の大会上位者同士を対戦させ、ログを収集（計10,000試合）



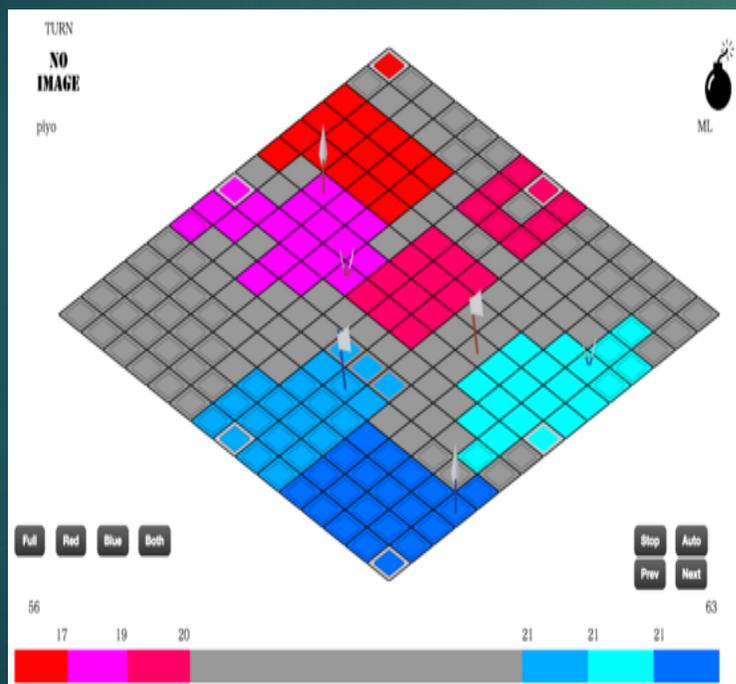
味方・敵の位置

隠伏・顕現状態

# 4：演習方法

## ▶ 1：ログデータ収集・整形

▶ 昨年の大会上位者同士を対戦させ、ログを収集（計10,000試合）



味方・敵の位置

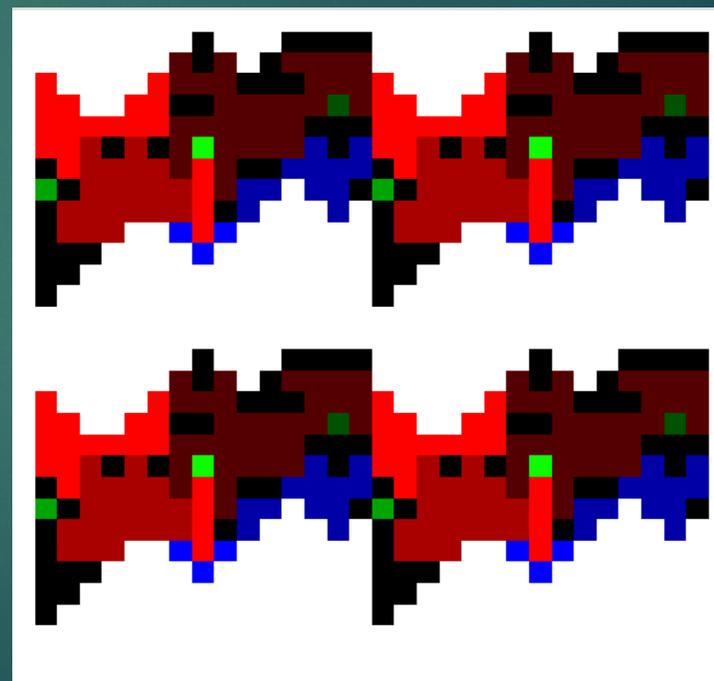
隠伏・顕現状態

盤面の状況

# 4 : 演習方法

- ▶ 1 : ログデータ収集・整形
- ▶ ログデータから学習用画像を生成 (各武器70,000枚)

```
# Turn information for samurai of side 0 with weapon 0
# <turn>
84
# <cure period>
0
# Samurai states: <position x> <position y> <hiding>
6 10 0
7 7 0
7 0 0
-1 -1 1
-1 -1 1
7 14 0
# Battle field states
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 2 8 2 8 8 8 8 9 9
9 9 9 9 0 8 2 2 2 8 8 9 9 9
9 9 9 9 9 8 2 2 2 4 9 9 9 9
9 9 9 9 9 8 2 2 2 2 9 9 9 9
9 9 9 9 1 8 2 2 2 2 9 9 9 9
9 9 9 1 1 1 2 2 2 2 4 9 9 9
9 9 1 1 1 1 2 2 2 3 3 3 9 9
9 9 9 1 1 1 1 0 0 0 0 4 9 9 9
9 9 0 1 1 1 1 0 0 0 0 9 9 9 9
9 0 0 8 1 1 1 1 1 3 3 3 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9
9 9 9 8 8 8 0 1 1 1 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9
```



# 4 : 演習方法

- ▶ 1 : ログデータ収集・整形
- ▶ 画像の生成には「OpenCV」を用いる
- ▶ OpenCV : Open Computer Vision Library
- ▶ Intelが開発した画像認識、解析、処理機能を持つオープンソフトウェア

# 4：演習方法

- ▶ 1：ログデータ収集・整形
- ▶ ログデータには、実際のゲームの中では得られない情報を含む  
例：視界外の盤面の塗り、隠伏状態の敵の位置
- ▶ 実戦形式に準じた、各ターンで受け取れる情報のみを抽出し、学習に用いる

# 4 : 演習方法

- ▶ 1 : ログデータ収集・整形
- ▶ ログデータから味方の位置、盤面の状況を抽出

```
# Turn information for samurai of side 0 with weapon 0
# <turn>
84
# <cure period>
0
# Samurai states: <position x> <position y> <hiding>
6 10 0
7 7 0
7 0 0
-1 -1
7 14 0
# Battle field states
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 2 8 2 8 8 8 8 9 9
9 9 9 9 0 8 2 2 2 8 8 9 9 9 9
9 9 9 9 9 8 2 2 2 4 9 9 9 9 9
9 9 9 9 9 8 2 2 2 2 9 9 9 9 9
9 9 9 9 1 8 2 2 2 2 2 9 9 9 9
9 9 9 1 1 1 2 2 2 2 2 4 9 9 9
9 9 1 1 1 1 2 2 2 3 3 3 3 9 9
9 9 9 1 1 1 1 0 0 0 0 4 9 9 9
9 9 0 1 1 1 1 0 0 0 0 9 9 9 9
9 0 0 8 1 1 1 1 1 3 3 3 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9 9
9 9 9 8 8 8 0 1 1 1 9 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9 9
```

```
6 10
7 7
7 0
```

```
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 2 8 2 8 8 8 8 9 9
9 9 9 9 0 8 2 2 2 8 8 9 9 9 9
9 9 9 9 9 8 2 2 2 4 9 9 9 9 9
9 9 9 9 9 8 2 2 2 2 9 9 9 9 9
9 9 9 9 1 8 2 2 2 2 2 9 9 9 9
9 9 9 1 1 1 2 2 2 2 2 4 9 9 9
9 9 1 1 1 1 2 2 2 3 3 3 3 9 9
9 9 9 1 1 1 1 0 0 0 0 4 9 9 9
9 9 0 1 1 1 1 0 0 0 0 9 9 9 9
9 0 0 8 1 1 1 1 1 3 3 3 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9 9
9 9 9 8 8 8 0 1 1 1 9 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9 9
```

# 4 : 演習方法

## ▶ 1 : ログデータ収集・整形

▶ 盤面の状況は各マスの塗りが数値（0～9）で記録されている

```
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 2 8 2 8 8 8 9 9 9
9 9 9 9 0 8 2 2 2 8 8 9 9 9 9
9 9 9 9 9 8 2 2 2 4 9 9 9 9 9
9 9 9 9 9 8 2 2 2 2 9 9 9 9 9
9 9 9 9 1 8 2 2 2 2 2 9 9 9 9
9 9 9 1 1 1 2 2 2 2 2 4 9 9 9
9 9 1 1 1 1 2 2 2 3 3 3 3 9 9
9 9 9 1 1 1 1 0 0 0 0 4 9 9 9
9 9 0 1 1 1 1 0 0 0 0 9 9 9 9
9 0 0 8 1 1 1 1 1 3 3 3 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9 9
9 9 9 8 8 8 0 1 1 1 9 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9 9
```

0～2 : チームAの武器ごとの塗り

3～5 : チームBの武器ごとの塗り

8 : 塗られていない

9 : 視界外の情報

# 4 : 演習方法

- ▶ 1 : ログデータ収集・整形
- ▶ OpenCVで画像に着色する各マスの色を決定

```
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 2 8 2 8 8 8 9 9 9
9 9 9 9 0 8 2 2 2 8 8 9 9 9 9
9 9 9 9 9 8 2 2 2 4 9 9 9 9 9
9 9 9 9 9 8 2 2 2 2 9 9 9 9 9
9 9 9 9 1 8 2 2 2 2 2 9 9 9 9
9 9 9 1 1 1 2 2 2 2 2 4 9 9 9
9 9 1 1 1 1 2 2 2 3 3 3 3 9 9
9 9 9 1 1 1 1 0 0 0 0 4 9 9 9
9 9 0 1 1 1 1 0 0 0 0 9 9 9 9
9 0 0 8 1 1 1 1 1 3 3 3 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9 9
9 9 9 8 8 8 0 1 1 1 9 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9 9
```

0 ~ 2 : 赤

3 ~ 5 : 青

8 : 黒

9 : 白

# 4 : 演習方法

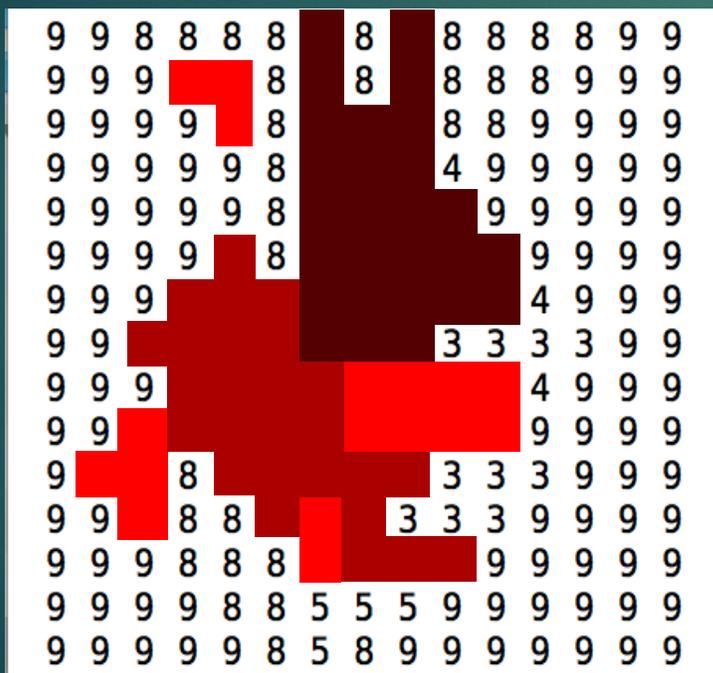
- ▶ 1 : ログデータ収集・整形
- ▶ OpenCVで15 × 15 の白画像を作成

```
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 2 8 2 8 8 8 9 9 9
9 9 9 9 0 8 2 2 2 8 8 9 9 9 9
9 9 9 9 9 8 2 2 2 4 9 9 9 9 9
9 9 9 9 9 8 2 2 2 2 9 9 9 9 9
9 9 9 9 1 8 2 2 2 2 2 9 9 9 9
9 9 9 1 1 1 2 2 2 2 2 4 9 9 9
9 9 1 1 1 1 2 2 2 3 3 3 3 9 9
9 9 9 1 1 1 1 0 0 0 0 4 9 9 9
9 9 0 1 1 1 1 0 0 0 0 9 9 9 9
9 0 0 8 1 1 1 1 1 3 3 3 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9 9
9 9 9 8 8 8 0 1 1 1 9 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9 9
```



# 4 : 演習方法

- ▶ 1 : ログデータ収集・整形
- ▶ OpenCVで各マスにある数値に従い、着色

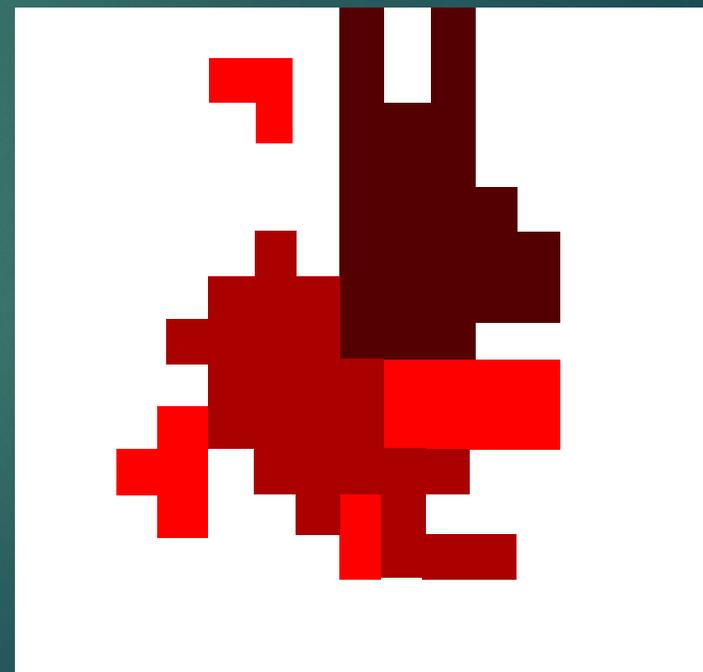


0 ~ 2 : 赤

3 ~ 5 : 青

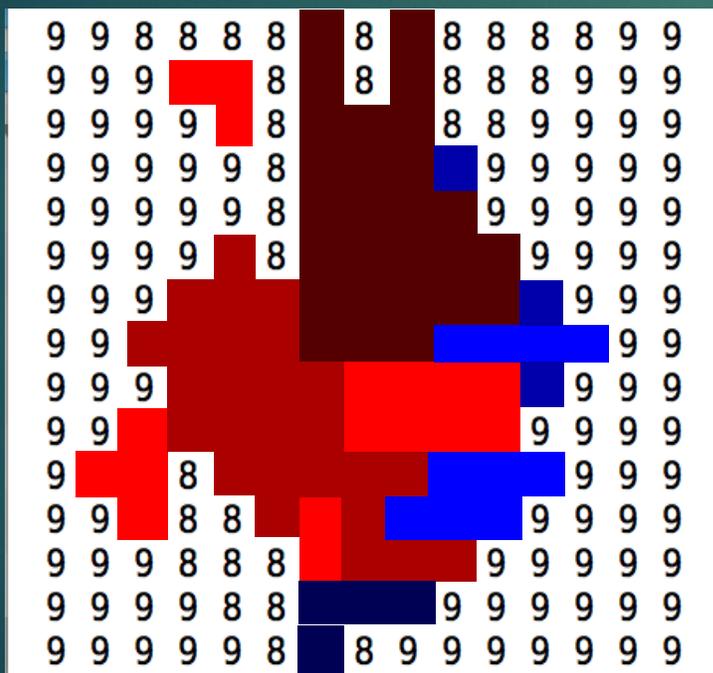
8 : 黒

9 : 白



# 4 : 演習方法

- ▶ 1 : ログデータ収集・整形
- ▶ OpenCVで各マスにある数値に従い、着色

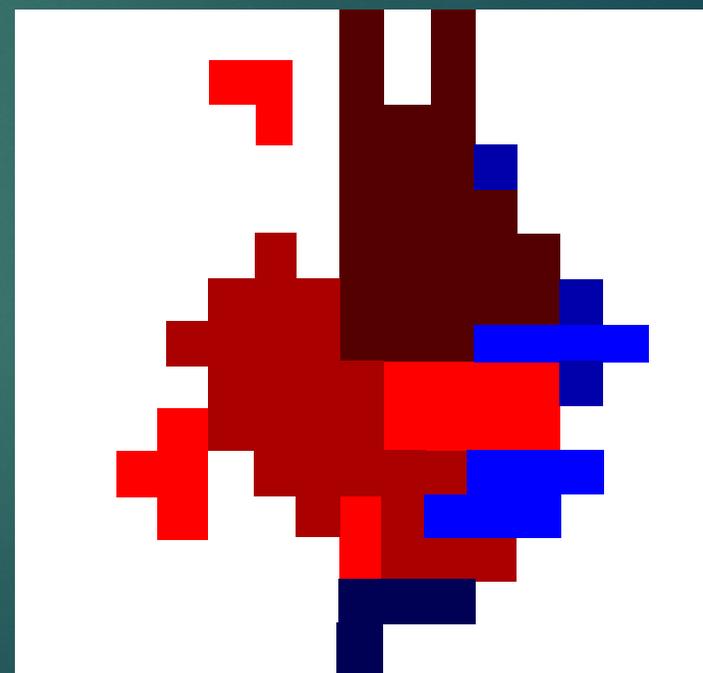


0 ~ 2 : 赤

3 ~ 5 : 青

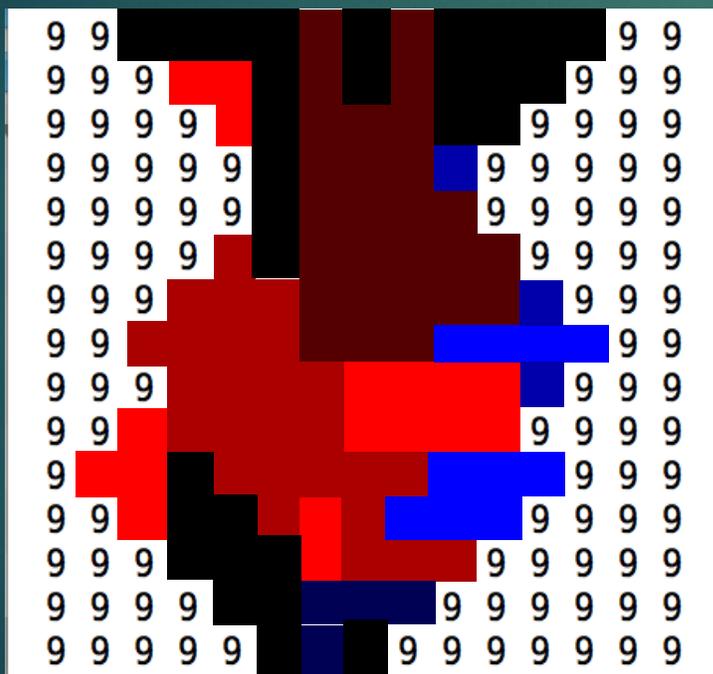
8 : 黒

9 : 白



# 4 : 演習方法

- ▶ 1 : ログデータ収集・整形
- ▶ OpenCVで各マスにある数値に従い、着色

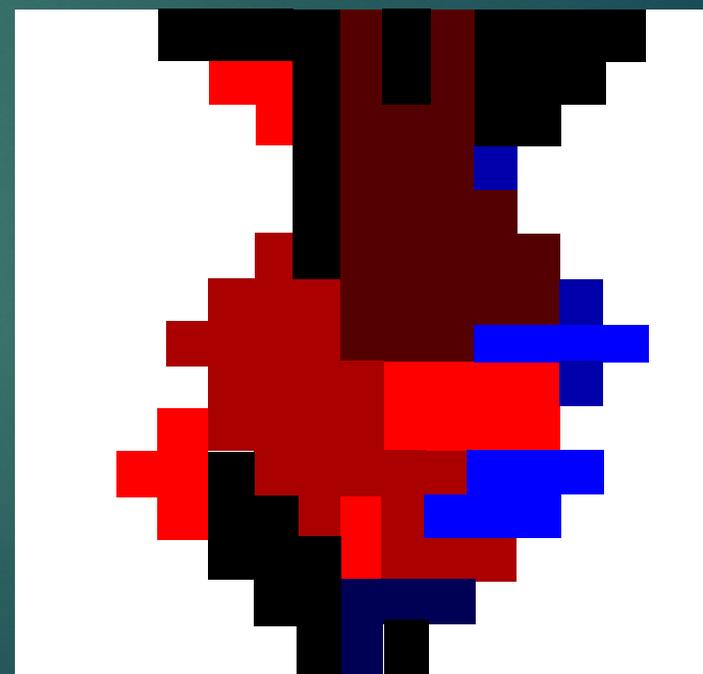


0 ~ 2 : 赤

3 ~ 5 : 青

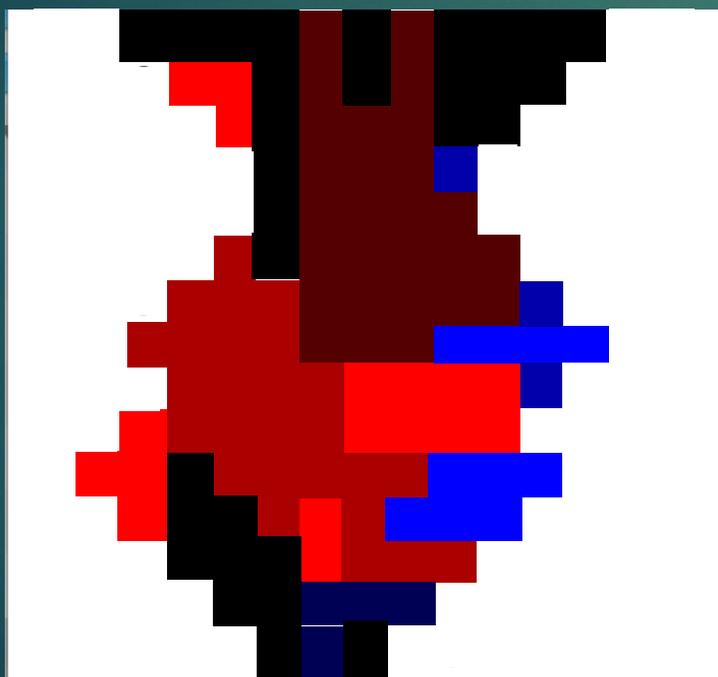
8 : 黒

9 : 白



# 4 : 演習方法

- ▶ 1 : ログデータ収集・整形
- ▶ OpenCVで各マスにある数値に従い、着色

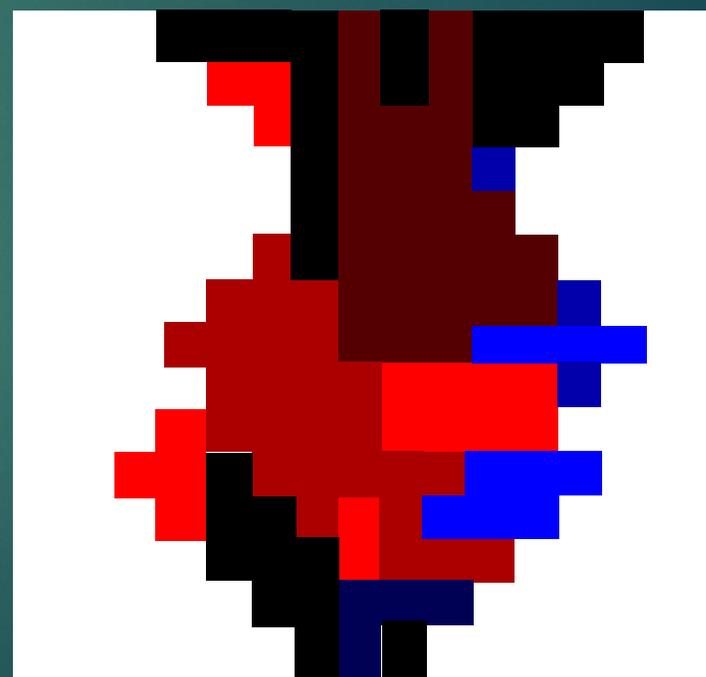


0 ~ 2 : 赤

3 ~ 5 : 青

8 : 黒

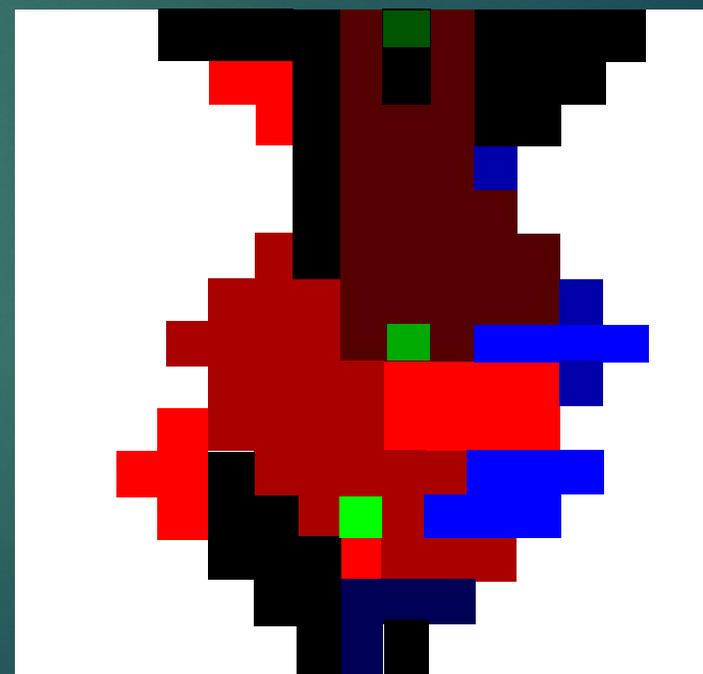
9 : 白



# 4 : 演習方法

- ▶ 1 : ログデータ収集・整形
- ▶ 味方の位置を緑で着色

6	10
7	7
7	0



# 4 : 演習方法

## ▶ 演習手順

- ▶ 1 : ログデータ収集・整形
- ▶ 2 : CNNを用いた教師あり学習
- ▶ 3 : 大会用プログラムに組み込み対戦

# 4 : 演習方法

- ▶ 2 : CNNを用いた教師あり学習
  - ▶ 教師あり学習 (Supervised Learning)
  - ▶ 事前に与えられた教師データをもとに学習を行う手法
- ▶ 教師データ
  - ▶ データとその正解ラベルをセットにしたもの

# 4 : 演習方法

## ▶ 2 : CNNを用いた教師あり学習

### ▶ 教師データ



# 4 : 演習方法

- ▶ 2 : CNNを用いた教師あり学習
  - ▶ 本演習における教師データ
  - ▶ 正解ラベルは盤面での実際にとった行動
  - ▶ 行動パターンは全部で879通り  70通り
  - ▶ 70クラスのカテゴリ

# 4 : 演習方法

- ▶ 2 : CNNを用いた教師あり学習
- ▶ 作成した画像データに正解ラベル（0～69）を付与
- ▶ 画像へのパス（スペース）正解ラベルとしてテキストに書き込み

```
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/52/5219.png 22  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/82/40918_2.png 65  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/1/12443.png 0  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/8/109936.png 63  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/38/76596_14.png 14  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/88/52368.png 68  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/77/21218.png 58  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/63/12823.png 42  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/88/28291.png 68  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/5/3374.png 20  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/555/47730.png 26  
/Users/g-2017/SamuraiCoding/traindata/first_1/0_1/557/3272_27.png 28
```

# 4 : 演習方法

- ▶ 2 : CNNを用いた教師あり学習
- ▶ 訓練用70000枚、テスト用35000枚
- ▶ 学習を行い正解率を評価

# 4 : 演習方法

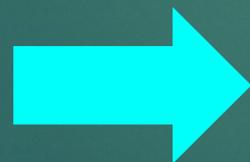
## ▶ 演習手順

- ▶ 1 : ログデータ収集・整形
- ▶ 2 : CNNを用いた教師あり学習
- ▶ 3 : 大会用プログラムに組み込み対戦

# 4 : 演習方法

- ▶ 3 : 大会用プログラムに組み込み対戦
- ▶ ゲーム情報を取得

```
# Turn information for samurai of side 0 with weapon 0
# <turn>
84
# <cure period>
0
# Samurai states: <position x> <position y> <hiding>
6 10 0
7 7 0
7 0 0
-1 -1 1
-1 -1 1
7 14 0
# Battle field states
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 2 8 2 8 8 8 9 9 9
9 9 9 9 0 8 2 2 2 8 8 9 9 9 9
9 9 9 9 9 8 2 2 2 4 9 9 9 9 9
9 9 9 9 9 8 2 2 2 2 9 9 9 9 9
9 9 9 9 1 8 2 2 2 2 2 9 9 9 9
9 9 9 1 1 1 2 2 2 2 2 4 9 9 9
9 9 1 1 1 1 2 2 2 3 3 3 9 9 9
9 9 9 1 1 1 1 0 0 0 0 4 9 9 9
9 9 0 1 1 1 1 0 0 0 0 9 9 9 9
9 0 0 8 1 1 1 1 1 3 3 3 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9 9
9 9 9 8 8 8 0 1 1 1 9 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9 9
```



CNNで分類



行動

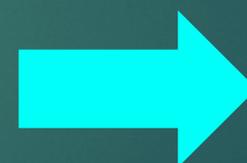
# 4 : 演習方法

- ▶ 3 : 大会用プログラムに組み込み対戦
- ▶ ゲーム情報から画像を生成

```
# Turn information for samurai of side 0 with weapon 0
# <turn>
84
# <cure period>
0
# Samurai states: <position x> <position y> <hiding>
6 10 0
7 7 0
7 0 0
-1 -1 1
-1 -1 1
7 14 0
# Battle field states
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 2 8 2 8 8 8 9 9 9
9 9 9 9 0 8 2 2 2 8 8 9 9 9 9
9 9 9 9 9 8 2 2 2 4 9 9 9 9 9
9 9 9 9 9 8 2 2 2 2 9 9 9 9 9
9 9 9 9 1 8 2 2 2 2 2 9 9 9 9
9 9 9 1 1 1 2 2 2 2 2 4 9 9 9
9 9 1 1 1 1 2 2 2 3 3 3 9 9 9
9 9 9 1 1 1 1 0 0 0 0 4 9 9 9
9 9 0 1 1 1 1 0 0 0 0 9 9 9 9
9 0 0 8 1 1 1 1 1 3 3 3 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9 9
9 9 9 8 8 8 0 1 1 1 9 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9 9
```



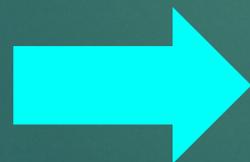
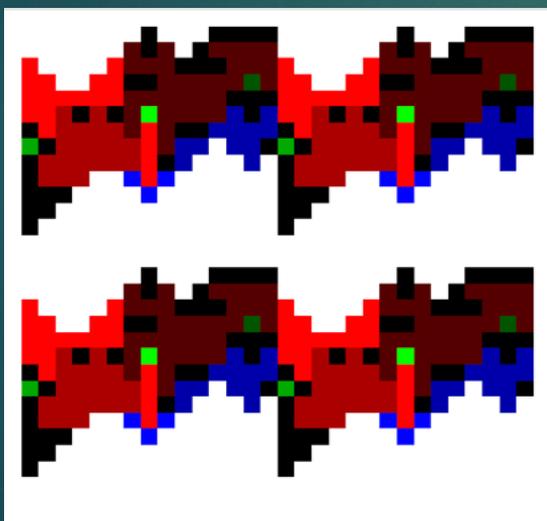
CNNで分類



行動

# 4 : 演習方法

- ▶ 3 : 大会用プログラムに組み込み対戦
- ▶ 画像をCNNで分類し、最も確率の高い手を次の手に選択



CNNで分類



行動

# 目次

- ▶ 1 : はじめに
- ▶ 2 : 機械学習におけるゲームAI
- ▶ 3 : 演習目的・内容
- ▶ 4 : 演習方法
- ▶ 5 : 演習結果・考察
- ▶ 6 : まとめ

# 5 : 演習結果・考察

## ▶ CNNによる教師あり学習

	槍	刀	鉞
正解率	40.04%	55.38%	25.9%
学習時間	2日	1日半	16時間
学習回数	150回	100回	50回

# 5 : 演習結果・考察

## ▶ CNNによる教師あり学習

### ▶ 学習回数の不足

 AlphaGoは50GPUで3週間、計3.4億回の学習を行っている

### ▶ 学習環境の調整の必要

### ▶ 同盤面で複数のラベルがある際に一つしか正解にならない

# 5 : 演習結果・考察

- ▶ 昨年大会上位者との対戦
- ▶ 多少動きはするが全く歯が立たない

# 5 : 演習結果・考察

- ▶ AlphaGoの仕組み
- ▶ 1 : 教師あり学習フェーズ
- ▶ 2 : 強化学習フェーズ
- ▶ 3 : 評価関数作成フェーズ
- ▶ 4 : モンテカルロ木探索フェーズ

打ち手を予測

勝てる手を予測

盤面を評価

数手先を読む

# 5 : 演習結果・考察

- ▶ AlphaGoの仕組み
- ▶ 1 : 教師あり学習フェーズ
- ▶ 2 : 強化学習フェーズ
- ▶ 3 : 評価関数作成フェーズ
- ▶ 4 : モンテカルロ木探索フェーズ

打ち手を予測

勝てる手を予測

盤面を評価

数手先を読む

# 5 : 演習結果・考察

- ▶ 昨年大会上位者との対戦
- ▶ 多少動きはするが全く歯が立たない
- ▶ 制限時間を超過

# 5 : 演習結果・考察

- ▶ 大会用プログラムに組み込み対戦
- ▶ ゲーム情報から画像を生成

```
# Turn information for samurai of side 0 with weapon 0
# <turn>
84
# <cure period>
0
# Samurai states: <position x> <position y> <health>
6 10 0
7 7 0
7 0 0
-1 -1 1
-1 -1 1
7 14 0
# Battle field states
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 8 2 8 2 8 8 8 9 9 9
9 9 9 9 0 8 8 2 2 2 8 8 9 9 9 9
9 9 9 9 9 8 8 2 2 2 4 9 9 9 9 9
9 9 9 9 9 8 8 2 2 2 2 9 9 9 9 9
9 9 9 9 1 8 2 2 2 2 2 2 9 9 9 9 9
9 9 9 1 1 1 2 2 2 2 2 4 9 9 9 9
9 9 1 1 1 1 2 2 2 3 3 3 9 9 9 9
9 9 9 1 1 1 1 0 0 0 0 4 9 9 9 9
9 9 9 0 1 1 1 1 0 0 0 0 9 9 9 9 9
9 0 0 8 1 1 1 1 1 3 3 3 9 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9 9 9
9 9 9 8 8 8 8 0 1 1 1 9 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9 9 9
```

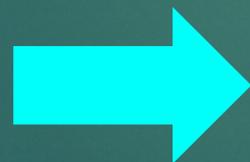
CNNで分類

行動

# 5 : 演習結果・考察

- ▶ 大会用プログラムに組み込み対戦
- ▶ ゲーム情報から学習結果を参照

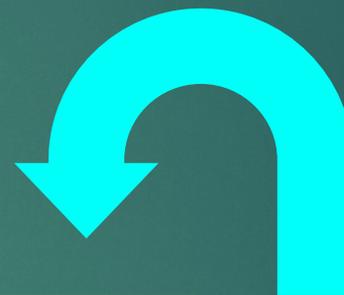
```
# Turn information for samurai of side 0 with weapon 0
# <turn>
84
# <cure period>
0
# Samurai states: <position x> <position y> <hiding>
6 10 0
7 7 0
7 0 0
-1 -1 1
-1 -1 1
7 14 0
# Battle field states
9 9 8 8 8 8 2 8 2 8 8 8 8 9 9
9 9 9 0 0 8 2 8 2 8 8 8 9 9 9
9 9 9 9 0 8 2 2 2 8 8 9 9 9 9
9 9 9 9 9 8 2 2 2 2 4 9 9 9 9 9
9 9 9 9 9 8 2 2 2 2 2 9 9 9 9 9
9 9 9 9 1 8 2 2 2 2 2 9 9 9 9 9
9 9 9 1 1 1 2 2 2 2 2 4 9 9 9 9
9 9 1 1 1 1 1 2 2 2 3 3 3 9 9 9
9 9 9 1 1 1 1 1 0 0 0 0 4 9 9 9
9 9 0 1 1 1 1 1 0 0 0 0 9 9 9 9
9 0 0 8 1 1 1 1 1 1 3 3 3 9 9 9 9
9 9 0 8 8 1 0 1 3 3 3 9 9 9 9 9
9 9 9 8 8 8 0 1 1 1 9 9 9 9 9 9
9 9 9 9 8 8 5 5 5 9 9 9 9 9 9 9
9 9 9 9 9 8 5 8 9 9 9 9 9 9 9 9
```



CNNで分類



行動



# 目次

- ▶ 1 : はじめに
- ▶ 2 : 機械学習におけるゲームAI
- ▶ 3 : 演習目的・内容
- ▶ 4 : 演習方法
- ▶ 5 : 演習結果・考察
- ▶ 6 : まとめ

# 6 : まとめ

- ▶ 本演習では「AlphaGo」の仕組みの一部を用いた教師あり学習を画像データを入力として行った
- ▶ 学習結果をもとに「SamurAI Coding」用プログラムを作成した

# 6 : まとめ

- ▶ 今後の課題
- ▶ 学習の入力形式の変更
- ▶ 強化学習の導入
- ▶ 学習環境の整備
- ▶ プログラムでのCNN呼び出し部分の時間短縮

# 参考文献

- ▶ ・画像
- ▶ [ICOON MONO](#)
- ▶ [DeepMind logo](#)
- ▶ ・AlphaGo
- ▶ [DeepMind](#)
- ▶ [Googleが出した囲碁ソフト「AlphaGo」の論文を翻訳して解説してみる。](#)
- ▶ [Google AlphaGoの仕組みを理解するーAI技術は今、どこまで進化したのか？](#)
- ▶ [AlphaGoの仕組み](#)
- ▶ [人間を超えたアルファ碁（AlphaGo）は、どのようにして強くなったのか](#)
- ▶ [囲碁AI”AlphaGo”はなぜ強いのか？](#)

# 参考文献

- ▶ ・ Samurai Coding
- ▶ [Samurai Coding 2016-17](#)
- ▶ ・ 深層学習
- ▶ [ニューラルネットワーク](#)
- ▶ [イラストで学ぶディープラーニング](#)
- ▶ [Convolutional Neural Networkとは何なのか](#)
- ▶ [Convolutional Neural Networkを実装する](#)