

# 「ソフトコンピューティング」(前半)

北海道大学 大学院情報科学研究科 山下 裕

2009 年後期

## ファジイ理論

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

## はじめに

- はじめに
- はじめに**
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

ソフトコンピューティング = L. A. Zadeh によって提唱された  
不精確性や不確実性を許容して過度な精密性の追求を避けること  
で、扱いやすさ・頑健性・低コスト性などを旨とする情報処理手法

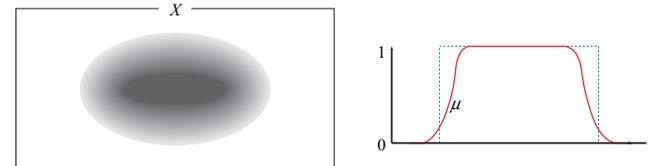
人によって何をソフトコンピューティングとするかは微妙に異なるが、  
次のような手法はソフトコンピューティングの一部として考えられている。

- ✓ ファジイ理論
- ✓ ニューラルネットワーク
- ✓ 遺伝的アルゴリズムなどの進化的計算
- ✓ 免疫アルゴリズム
- ✓ 強化学習
- ✓ サポートベクタマシン (人によっては含めないかも)
- ✓ カオスを用いた手法

## ファジイ集合

- はじめに
- ファジイ理論**
- ファジイ集合**
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

ファジイ集合 (Fuzzy set): あいまいな (= fuzzy な) 集合のこと。  
全体集合  $X$  のファジイ部分集合とは、  
$$\mu : X \rightarrow [0, 1]$$
  
のことで、関数  $\mu$  はメンバーシップ関数と呼ばれる。



- 1 ... その要素がファジイ集合に含まれている
- 0 ... その要素がファジイ集合に含まれていない

## ファジイラベル

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

通常、メンバーシップ関数には、“名前”がついており、これを**ファジイラベル**という。

たとえば、通常の集合では「100 以上の数」というのに対して、「大きな数」といった名前がファジイラベルである。

ファジイラベル  $A$  に対応したメンバーシップ関数を  $\mu_A$  で表わす。また、ファジイラベルとファジイ集合はしばしば同一視される。

## ファジイ集合の表記

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

有限集合の場合:  
メンバーシップ関数  $\mu_S$  を持つファジイ集合  $S$  を

$$S = \mu_S(x_1)/x_1 + \cdots + \mu_S(x_n)/x_n = \sum_{i=1}^n \mu_S(x_i)/x_i$$

と書く。  
 $/, +$  は、足し算・割り算の意味ではなく、単なる記号である。

無限集合の場合:  
メンバーシップ関数  $\mu_S$  を持つファジイ集合  $S$  を

$$S = \int_X \mu_S(x)/x$$

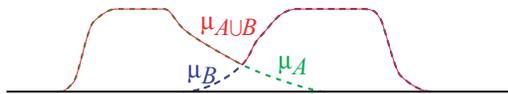
と書く。これも単なる約束事である。

## ファジイ集合の演算 (min-max 法) (1)

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

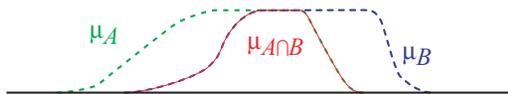
集合和:

$$A \cup B = \int \mu_A(x) \vee \mu_B(x) / x = \int \max(\mu_A(x), \mu_B(x)) / x$$



集合積:

$$A \cap B = \int \mu_A(x) \wedge \mu_B(x) / x = \int \min(\mu_A(x), \mu_B(x)) / x$$



## ファジイ集合の演算 (min-max 法) (2)

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

補集合:

$$\bar{A} = \int \overline{\mu_A(x)} / x = \int 1 - \mu_A(x) / x$$

包含関係:

$$A \subset B : \mu_A(x) \leq \mu_B(x), \forall x$$

相当関係:

$$A = B : \mu_A = \mu_B$$

その他、“代数積-代数和法”もある。

$$A \cup B : \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$$

$$A \cap B : \mu_A(x)\mu_B(x)$$

## ファジイ集合演算の性質

- はじめに
- ファジイ理論
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

反射率  $A \subset A$   
 反対称律  $A \subset B, B \subset A \rightarrow A = B$   
 推移率  $A \subset B, B \subset C \rightarrow A \subset C$   
 べき等律  $A \cup A = A, A \cap A = A$   
 交換律  $A \cup B = B \cup A, A \cap B = B \cap A$   
 結合律  $(A \cup B) \cup C = A \cup (B \cup C), (A \cap B) \cap C = A \cap (B \cap C)$   
 吸収律  $A \cup (A \cap B) = A, A \cap (A \cup B) = A \cap B$   
 分配律  $A \cup (B \cap C) = (A \cup B) \cap (A \cup C), A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$   
 乗数法則  $A \cup X = X, A \cap X = A, A \cap \phi = \phi, A \cup \phi = A$   
 二重否定の法則  $\bar{\bar{A}} = A$   
 ド・モルガンの法則  $\overline{A \cup B} = \bar{A} \bar{B}, \overline{A \cap B} = \bar{A} \cup \bar{B}$

成り立たない法則:  
 相補律 「 $A \cup \bar{A} = X, A \cap \bar{A} = \phi$ 」は一般に成り立たない。

## ファジイ関係

- はじめに
- ファジイ理論
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

通常の集合でまず考えよう。

たとえば、集合  $X = \{1, 2, 3, 4, 5\}, Y = \{1, 2, 3\}$  において、  
 「 $\sim$ より小さい」:  $x < y (x \in X, y \in Y)$   
 という“関係”は、直積集合  $X \times Y$  上での部分集合、  

$$R = \{(1, 2), (1, 3), (2, 3)\}$$
  
 によって表わされる。

では、ファジイな関係とはなんであろうか?

$X$  から  $Y$  へのファジイ関係は、  

$$\mu_R : X \times Y \rightarrow [0, 1]$$
  
 で表わされる。すなわち、 $X \times Y$  上でのファジイ集合で表わされる。

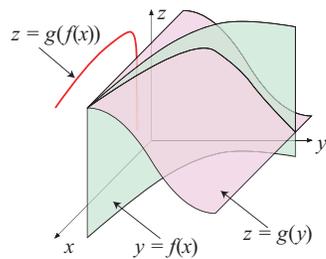
## ファジイ関係の合成 (1)

- はじめに
- ファジイ理論
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

通常関数  $f : X \rightarrow Y, g : Y \rightarrow Z$  の合成関数、

$$g \circ f = g(f(x))$$

のグラフは、 $f$  のグラフと  $Z$  の直積集合、 $g$  のグラフと  $X$  の直積集合の交点集合を  $Y$  に沿って  $X \times Z$  に射影したものである。



## ファジイ関係の合成 (2)

- はじめに
- ファジイ理論
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

$R$  を  $X \times Y$  のファジイ関係、 $S$  を  $Y \times Z$  のファジイ関係とすると、  

$$R \circ S = \int \mu_{R \circ S}(x, z) = \int \bigvee_y \{ \mu_R(x, y) \wedge \mu_S(y, z) \} / (x, z)$$
  
 をファジイ関係の合成という。

(注意) ファジイ集合論における  $\circ$  は通常と逆の順番に書く。

min-max 法では、

$$\mu_{R \circ S} = \max_y \{ \min(\mu_R(x, y), \mu_S(y, z)) \}$$

となる。

## ファジイ関係によるファジイ集合の写像

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

$A \subset X$  をファジイ集合、 $R$  を  $X \times Y$  のファジイ関係とすると、 $A$  を  $Y$  上に写像したものは、

$$A \circ R = \int \mu_{A \circ R} / y = \int \bigvee_x \{ \mu_A(x) \wedge \mu_R(x, y) \} / y$$

と定義され、 $Y$  上の新たなファジイ集合となる。

(注意) これも、 $\circ$  は通常と逆の順番に書くようである。

min-max 法では、

$$\mu_{A \circ R} = \max_x \{ \min(\mu_A(x), \mu_R(x, y)) \}$$

となる。

## ファジイルール

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

ファジイルール:

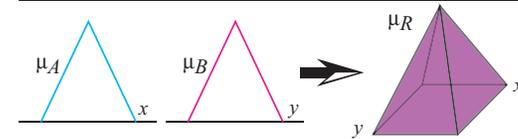
if  $x$  is  $A$  then  $y$  is  $B$        $x$  is  $A$ : 前件部  
 $y$  is  $B$ : 後件部

これを、ファジイ関係で表わす。

$A \subset X, B \subset Y$  に対して、直積、

$$R = A \times B \\ = \int \mu_R(x, y) / (x, y) = \int \mu_A(x) \wedge \mu_B(y) / (x, y) \quad (C \ X \times Y)$$

は、上記のファジイルールを表現するファジイ関係である。



## ファジイ推論

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

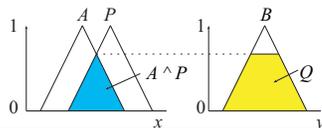
「 $x$  が  $A$  ならば」というルールに部分的にマッチしているファジイ集合  $P$  が与えられたときに、それはどのように推論されるであろうか?

→ **ファジイ関係によるファジイ集合の写像を使う**

ファジイルール "if  $x$  is  $A$  then  $y$  is  $B$ " の  $x$  にファジイ集合  $P$  を与えたときのファジイ推論:

$$Q = P \circ R$$

$$\mu_Q = \bigvee_x [\mu_P(x) \wedge \{ \mu_A(x) \wedge \mu_B(y) \}] = \bigvee_x \{ [\mu_P(x) \wedge \mu_A(x)] \wedge \mu_B(y) \} \\ = [\bigvee_x \{ \mu_P(x) \wedge \mu_A(x) \}] \wedge \mu_B(y)$$



## 非ファジイな事実からのファジイ推論

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

ファジイルールに対しファジイでない命題  $x = x_0$  を与えたとき:

$$\mu_P(x) = \begin{cases} 1, & x = x_0 \\ 0, & x \neq x_0 \end{cases}$$

を代入すると、

$$\mu_Q = \mu_A(x_0) \wedge \mu_B(y)$$

## 多重ファジイ推論

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論**
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

複数の前件部に部分的にマッチした場合はどうなるのでしょうか？

ファジイルール: if  $x$  is  $A_1$  then  $y$  is  $B_1$ , if  $x$  is  $A_2$  then  $y$  is  $B_2, \dots$

に対する推論結果:

$$\begin{aligned} \mu_Q &= \bigvee_i \bigvee_x [\mu_{P_i}(x) \wedge \{\mu_{A_i}(x) \wedge \mu_{B_i}(y)\}] \\ &= \bigvee_i \bigvee_x \{\mu_{P_i}(x) \wedge \mu_{A_i}(x)\} \wedge \mu_{B_i}(y) \end{aligned}$$

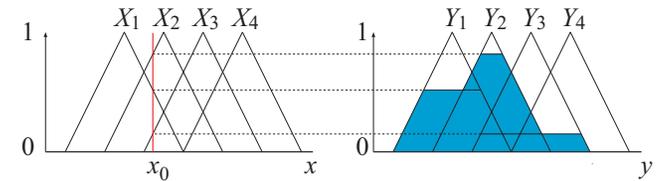
$$Q = \int \mu_Q / y = \sum_i \int_x [\bigvee_x \{\mu_{P_i}(x) \wedge \mu_{A_i}(x)\} \wedge \mu_{B_i}(y) / y]$$

ルール同士は OR 結合

## 多重ファジイ推論の例

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論**
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

if  $x$  is  $X_1$  then  $y$  is  $Y_1$ , if  $x$  is  $X_2$  then  $y$  is  $Y_2$ ,  
if  $x$  is  $X_3$  then  $y$  is  $Y_3$ , if  $x$  is  $X_4$  then  $y$  is  $Y_4$



## 非ファジイ化

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化**
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

実際にファジイ推論を用いるときは、ファジイ推論の結果得られたファジイ集合  $Q$  を通常の数値に直さなくてはならない。この操作を**非ファジイ化** (defuzzification) という。

重心法による非ファジイ化:  
ファジイ集合  $Q$  を非ファジイ化した値は、

$$y_0 = \frac{\int y \mu_Q(y) dy}{\int \mu_Q(y) dy}$$

1 次モーメントを 0 次モーメントで割る = 重心を求める操作

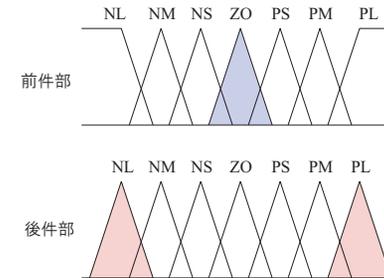
## 通常の推論との違い

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い**
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- ✓ ルール同士が AND ではなく OR 結合
- ✓ "if  $A$  then  $B$ " が、 $\bar{A} \vee B$  でなく、 $A \wedge B$

その結果、矛盾するルールでも推論結果が得られることがある。

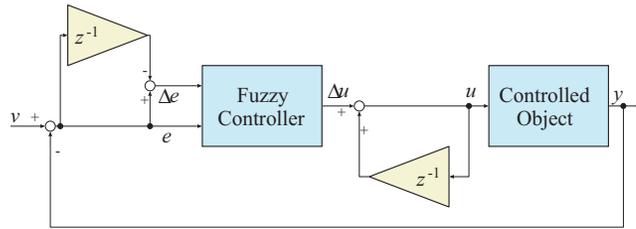
【例】 「正面に障害物が見えたら、右か左に大きくハンドルを切る」  
if  $x$  is ZO then  $y$  is NL, および if  $x$  is ZO then  $y$  is PL



もっとも、推論結果を重心法で非ファジイ化すると、障害物にぶつかってしまうが...

## ファジイ制御の例 (1)

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

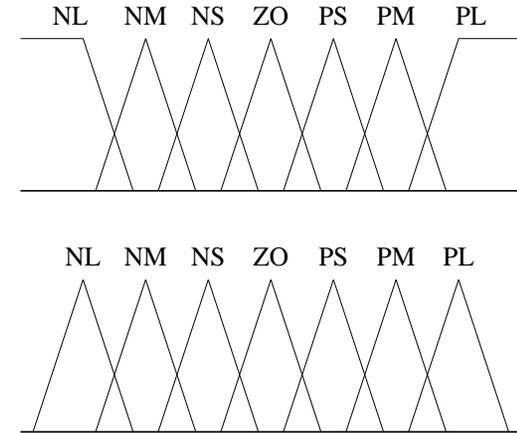


$z^{-1}$  は時間遅れ要素

## ファジイ制御の例 (2)

- はじめに
- ファジイ理論**
- ファジイ集合
- ファジイ集合の表記
- ファジイ演算
- ファジイ関係
- ファジイ関係の合成
- ファジイルール
- ファジイ推論
- 非ファジイ化
- 通常の推論との違い
- ファジイ制御の例
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

前件部・後件部のメンバーシップ関数



## ニューラルネットワーク

- はじめに
- ファジイ理論
- ニューラルネットワーク**
- NN とは
- 神経細胞
- 神経細胞のモデル
- Hebb 則
- Hebb 則の意味
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

## ニューラルネットワークとは

ニューラルネットワーク (Neural network) とは、  
脳機能の (一部の) 特性を計算によって表現することを目指した数  
学モデル

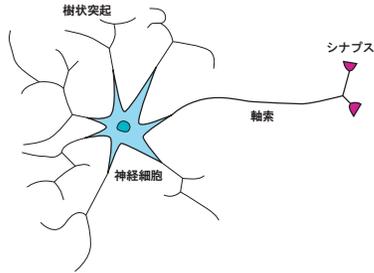
何に使えるか?

- ✓ 関数学習
- ✓ 最適化
- ✓ 連想記憶
- ✓ クラスタリング
- ✓ カオスダイナミクスの実現
- ✓ その他...

## 神経細胞

はじめに  
 ファジ理論  
 ニューラルネットワーク  
 NN とは  
 神経細胞  
 神経細胞のモデル  
 Hebb 則  
 Hebb 則の意味  
 FF 型 NN  
 SA とボルツマンマシン  
 Hopfield NN と連想記憶  
 自己組織化ネットワーク  
 遺伝的アルゴリズム  
 遺伝的プログラミング  
 多目的最適化と GA  
 進化戦略

典型的な神経細胞:



- ✓ 樹状突起: 神経細胞の周りに発達する突起。他の神経細胞からの信号を受け取る部分。
- ✓ 軸索: 神経細胞から長く伸びる部分。他の神経細胞に信号を伝える。
- ✓ シナプス: 軸索先端と樹状突起の間で形成される、信号の受け渡し部分。電気的なシナプスもあるが、多くは、化学物質による伝達。

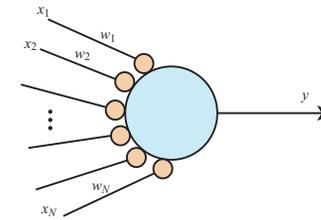
ソフトコンピューティング

2009 年後期 - 25 / 115

## 神経細胞のモデル

はじめに  
 ファジ理論  
 ニューラルネットワーク  
 NN とは  
 神経細胞  
 神経細胞のモデル  
 Hebb 則  
 Hebb 則の意味  
 FF 型 NN  
 SA とボルツマンマシン  
 Hopfield NN と連想記憶  
 自己組織化ネットワーク  
 遺伝的アルゴリズム  
 遺伝的プログラミング  
 多目的最適化と GA  
 進化戦略

実際の神経細胞のモデル化というより、役に立つ神経細胞モデル



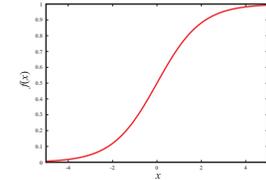
ニューロンのモデル:

$$y = f\left(\sum_{i=1}^N w_i x_i + \theta\right)$$

$f(\cdot)$ : シグモイド関数  
 $\theta$ : オフセット  
 $w_i$ : 重み

シグモイド関数:

$$f(x) = \frac{1}{1 + e^{-kx}} \quad (k > 0)$$



ソフトコンピューティング

2009 年後期 - 26 / 115

## Hebb 則 (1)

はじめに  
 ファジ理論  
 ニューラルネットワーク  
 NN とは  
 神経細胞  
 神経細胞のモデル  
 Hebb 則  
 Hebb 則の意味  
 FF 型 NN  
 SA とボルツマンマシン  
 Hopfield NN と連想記憶  
 自己組織化ネットワーク  
 遺伝的アルゴリズム  
 遺伝的プログラミング  
 多目的最適化と GA  
 進化戦略

$y = \phi(x)$  なる関数を学習したい。

正しい  $(x, y)$  のペアの列が与えられたとする。そのときの  $y$  を (ニューロンの実際の出力と区別して)、 $\bar{y}$  と書き、「教師信号」とよぶ。これらより、重み係数  $w_i$  とオフセットを「学習」したい。

Hebb 則:

$$w_{i,new} = w_{i,old} - \eta(y - \bar{y})f'(\sum w_i x_i + \theta)x_i$$

$$\theta_{new} = \theta_{old} - \eta(y - \bar{y})f'(\sum w_i x_i + \theta)$$

ソフトコンピューティング

2009 年後期 - 27 / 115

## Hebb 則 (2)

はじめに  
 ファジ理論  
 ニューラルネットワーク  
 NN とは  
 神経細胞  
 神経細胞のモデル  
 Hebb 則  
 Hebb 則の意味  
 FF 型 NN  
 SA とボルツマンマシン  
 Hopfield NN と連想記憶  
 自己組織化ネットワーク  
 遺伝的アルゴリズム  
 遺伝的プログラミング  
 多目的最適化と GA  
 進化戦略

シグモイド関数の微分:

$$f'(x) = \frac{ke^{-kx}}{(1 + e^{-kx})^2} = kf(x)(1 - f(x))$$

よって、Hebb 則は、以下のように書ける。

$$w_{i,new} = w_{i,old} - \bar{\eta}(y - \bar{y})y(1 - y)x_i$$

$$\theta_{new} = \theta_{old} - \bar{\eta}(y - \bar{y})y(1 - y)$$

ただし、 $\bar{\eta} = k\eta$

ソフトコンピューティング

2009 年後期 - 28 / 115

## Hebb 則の意味

- はじめに
- ファジイ理論
- ニューラルネットワーク
- NN とは
- 神経細胞
- 神経細胞のモデル
- Hebb 則
- Hebb 則の意味
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

評価関数:

$$E = \frac{1}{2}(y - \bar{y})^2 \rightarrow \min$$

$$\mathbf{w} = (w_1, \dots, w_N), \quad \mathbf{x} = (x_1, \dots, x_N)^T$$

とすると、

$$\left(\frac{\partial E}{\partial \mathbf{w}}\right)^T = (y - \bar{y})f'(\mathbf{w}\mathbf{x} + \theta)\mathbf{x}^T, \quad \frac{\partial E}{\partial \theta} = (y - \bar{y})f'(\mathbf{w}\mathbf{x} + \theta)$$

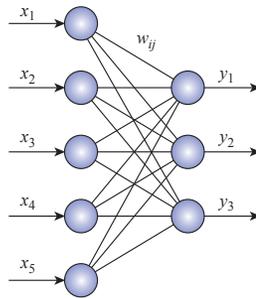
したがって、Hebb 則は以下のように書ける。

$$w_{i,new} = w_{i,old} - \eta \left(\frac{\partial E}{\partial w}\right)^T, \quad \theta_{new} = \theta_{old} - \eta \frac{\partial E}{\partial \theta}$$

つまり、Hebb 則は評価関数  $E$  に対する最急降下法。

## パーセプトロン

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略



- ✓ 重み  $w_{ij}$  とオフセット  $\theta_j$  を Hebb 則で学習
- ✓ 出力層が 1 つのニューロンの場合、単純パーセプトロンと呼ばれる。

## フィードフォワード型ニューラルネットワーク

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

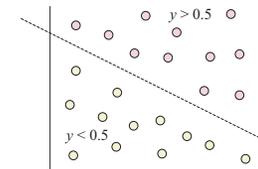
## パーセプトロンの限界 (1)

パーセプトロンの出力:

$$y = f\left(\sum_{i=1}^N w_i x_i + \theta\right)$$

が 0.5 を超える領域は、

$$\left\{ x \mid \sum_{i=1}^N w_i x_i + \theta > 0 \right\}$$



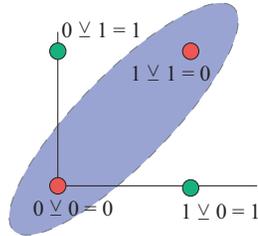
つまり線形分離できる関数しか学習できない

## パーセプトロンの限界 (2)

- はじめに
- フジイ理論
- ニューラルネットワーク
- FF 型 NN
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

線形分離できない関数の例: XOR 関数

$$0 \vee 0 = 0, 0 \vee 1 = 1, 1 \vee 0 = 1, 1 \vee 1 = 0$$

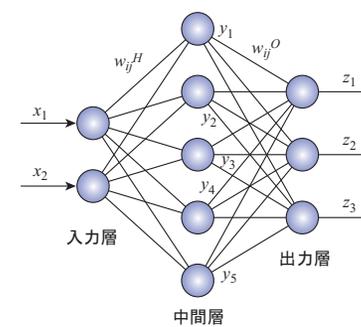


このような関数はパーセプトロンでは実現できない  
 では、複数のパーセプトロンの出力の和では実現できないだろうか?  
 → 階層型ニューラルネットワーク

## 階層型ニューラルネットワーク

- はじめに
- フジイ理論
- ニューラルネットワーク
- FF 型 NN
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

3層ニューラルネットワークの例:



$$z_i = g \left( \sum_{j=1}^M w_{i,j}^O y_j + \theta_i^O \right)$$

$$y_j = f \left( \sum_{k=1}^N w_{j,k}^H x_k + \theta_j^H \right)$$

行列表現では、

$$z = g(W^O f(W^H x + \theta^H) + \theta^O)$$

- ✓  $f(\cdot)$  は、シグモイド関数であることが多い。
- ✓  $g(\cdot)$  は、恒等写像かシグモイド関数であることが多い。出力の範囲を  $(0, 1)$  に制限したいときにはシグモイド関数を使う。

## 階層型ニューラルネットワークの表現能力

- はじめに
- フジイ理論
- ニューラルネットワーク
- FF 型 NN
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

階層型ニューラルネットワークには、パーセプトロンのような制限がないことが示される。

**Funahashi の定理:**  $N$  次元実空間内の有界集合  $S$  上で定義された任意の連続関数  $z = \phi(x)$  に対し、3層型ニューラルネットワークは、中間層のニューロンを増やすことによって、写像  $z = \phi(x)$  を任意の精度で近似できる。すなわち、任意の  $\epsilon (> 0)$  に対し、

$$\left| z - g(W^O f(W^H x + \theta^H) + \theta^O) \right| < \epsilon$$

となる、 $M, W^H, W^O, \theta^H, \theta^O$  が存在する。

K.Funahashi: On the approximation realization of continuous mappings by neural networks, *Neural Networks*, 2(3), 183-192 (1989)

## オフセットの取り扱い

- はじめに
- フジイ理論
- ニューラルネットワーク
- FF 型 NN
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

入力層および中間層において、常に 1 という値を持つニューロンを付加する。

$$x_{N+1} = 1, \quad y_{M+1} = 1$$

そして、

$$w_{i,M+1}^O = \theta_i^O, \quad w_{j,N+1}^H = \theta_j^H$$

とおくと、

$$z_i = g \left( \sum_{j=1}^{M+1} w_{i,j}^O y_j \right), \quad y_j = f \left( \sum_{k=1}^{N+1} w_{j,k}^H x_k \right)$$

のように簡単に書ける。

## 階層型ニューラルネットワークの学習 (1)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN**
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

階層型ニューラルネットワークの学習において、教師信号は出力層のみに与えられる。つまり、単純な Hebb 則は中間層=出力層間の結合重みの学習にしか適用できない。

出力層の学習: (Hebb 則)

$$\widehat{W}_{new}^O = \widehat{W}_{old}^O - \eta^O \text{diag}(e^O) g'(\widehat{W}^O \hat{y}) \hat{y}^T$$

ただし、 $\text{diag}(\cdot)$  は引数ベクトルの各要素を対角成分に持つ対角行列で、

$$\widehat{W}^O = [\mathbf{W}^O \quad \boldsymbol{\theta}^O], \quad \hat{y} = (y_1, \dots, y_M, 1)^T, \\ e^O = \mathbf{z} - \bar{\mathbf{z}} \quad \dots \text{(誤差信号, } \bar{\mathbf{z}} \text{ は教師信号)}$$

また、 $g'(\cdot)$  は引数の各成分に  $g'(\cdot)$  を適用してできたベクトルを表すとする。

中間層のニューロンに対しても、誤差信号に相当するものを生成する必要がある。

## 階層型ニューラルネットワークの学習 (2)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN**
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

Rumelhart らの誤差逆伝播法 (Backpropagation): 中間層ニューロンに対する誤差信号:

$$e^H = \mathbf{W}^O \text{diag}(e^O) g'(\widehat{W}^O \hat{y})$$

入力層=中間層間重みの更新: ( $e^H$  を用いた Hebb 則)

$$\widehat{W}_{new}^H = \widehat{W}_{old}^H - \eta^H \text{diag}(e^H) f'(\widehat{W}^H \hat{x}) \hat{x}^T$$

## 誤差逆伝播法の意味

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN**
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

実は、誤差逆伝播法による学習も、評価関数

$$E = \frac{1}{2} (\mathbf{z} - \bar{\mathbf{z}})^T (\mathbf{z} - \bar{\mathbf{z}}) = \frac{1}{2} e^{OT} e^O \rightarrow \min$$

に対する最急降下法。

$$\widehat{W}_{new}^O = \widehat{W}_{old}^O - \eta^O \left( \frac{\partial E}{\partial \widehat{W}^O} \right)^T = \widehat{W}_{old}^O - \eta^O \left( \frac{\partial E}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \widehat{W}^O} \right)^T$$

$$\widehat{W}_{new}^H = \widehat{W}_{old}^H - \eta^H \left( \frac{\partial E}{\partial \widehat{W}^H} \right)^T = \widehat{W}_{old}^H - \eta^H \left( \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \widehat{W}^H} \right)^T$$

ここで、 $\frac{\partial E}{\partial \mathbf{z}} = e^{OT}$ ,  $\frac{\partial E}{\partial \hat{y}} = e^{HT}$ 。つまり、誤差信号として  $e^H$  を用いればよい。これらの式は、実際は行列では表すことができずテンソルが必要。転置は縦横を入れ替えるぐらいの意味と考えて欲しい。

## 慣性項の付加

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN**
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

誤差逆伝播法による学習は、“その学習データの下での” 最急降下方向に、重みを修正する方法。  
本来は、**全ての学習データの下での**評価関数の総和 (あるいは最大値) の**最急降下法が望ましい**。

そこで、過去の修正を、引き続き加え続ける方法が提案された。

$$\widehat{W}_{(k+1)}^{[O,H]} = \widehat{W}_{(k)}^{[O,H]} + \Delta \widehat{W}_{(k)}^{[O,H]} \\ \Delta \widehat{W}_{(k)}^{[O,H]} = -\eta^{[O,H]} \left( \frac{\partial E}{\partial \widehat{W}^{[O,H]}} \right)^T + \alpha^{[O,H]} \Delta \widehat{W}_{(k-1)}^{[O,H]}$$

- ✓ 水色の項 … 誤差逆伝播法による修正項
- ✓ 赤色の項 … 慣性項

# 過学習

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- パーセプトロン
- パーセプトロンの限界
- 階層型 NN
- 階層型 NN の表現能力
- オフセットの取り扱い
- 階層型 NN の学習
- 誤差逆伝播法の意味
- 慣性項の付加
- 過学習
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- ✓ 階層型ニューラルネットワークは入出力データの組を学習し、その写像の近似を生成することができる。
- ✓ 与えられた入出力データ以外の点においても、適当に補間した値を出力してくれることが期待できる。(汎化)

学習の仕方によっては、汎化がうまくいかないことがある。  
たとえば...

- ✓ 中間ニューロン数が比較的多いにもかかわらず、少ない入出力データの組で繰り返し学習した場合。
- ✓ 入出力データに比較的大きめのノイズが乗っているにもかかわらず、少ない入出力データの組で繰り返し学習した場合。

少ない入出力データの組で繰り返し学習した場合、別の検証用データでの誤差は、始めは減少するが、その後増加していくことがある。  
⇒ 過学習

# 焼きなまし法とボルツマンマシン

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

# 有限の状態数の場合の最適化問題

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

ここでは、有限の選択肢の中から、ある評価関数を最適化するものを選ぶ問題を考える。

[例 1] たとえば

$$x = (0, 1, 1, 0, 0, 1, 1, 0, 1, \dots, 0, 1)$$

のような、有限長のビット列に対し、評価関数:  $F(x)$  を最小化するような最適化問題を考えることができる。

[例 2] たとえば、アルファベット A,B,C,...,X,Y,Z を任意の順番で、並び替えた  $x = (J,F,L,\dots,K,Q,D,E)$  のようなものの中から、最適なものを見つけ出す。

# 巡回セールスマン問題

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

巡回セールスマン問題 (TSP, Traveling Salesman Problem) とは? いくつかの都市があり、それらの都市の間の移動距離が与えられているものとする。セールスマンは全ての都市を 1 回ずつ訪問し、最後は元の都市に戻るとして、その合計移動距離を最小化するような、訪問順はどのようなものか?

前ページ [例 2] のような定式化をして、

$$x = (x_1, \dots, x_N)$$

とおく。ただし、 $x_i$  は  $i$  番目に訪問する都市の番号を表し、 $x_i \neq x_j$  ( $i \neq j$ ) である。そのとき、

$$F(x) = \sum_{i=1}^N d(x_i, x_{i+1}) \quad (\text{ただし、} x_{N+1} = x_1)$$

を最小化する問題。

$d(x_i, x_j)$  は  $x_i$  の都市と  $x_j$  の都市の間の移動距離

## 焼きなまし法とは

- はじめに
- ファジ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

焼きなまし法 (SA, Simulated Annealing) とは、

- ✓ このような大域的最適化問題に対する、確率的メタアルゴリズム。
- ✓ 大域的最適解に対して、よい近似を与える。
- ✓ S.Kirkpatrick, C.D.Gelatt, M.P.Vecchi ら (1983) および V.Cerny(1985)
- ✓ 金属工学における“焼きなまし”から来ている。

最初は温度が高い = 大きく動き局所的な極小値から脱出、広く探査

徐々に最初を下げる = より良い極小値に収束



## 状態近傍

- はじめに
- ファジ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

基本的には確率的な探索だが、闇雲に探索しても意味はない。  
現在の状態の「近く」に動いて探索 = 徐々に良い値を得る

探索空間を  $S$  とおく。  
ある状態  $x \in S$  に対し、その状態近傍  $N(x)$  を、 $S$  の要素からなる集合として与える。

仮定 1:  $x \notin N(x)$   
仮定 2: 任意の 1 つの状態からその「状態近傍」, 「状態近傍のある点に対する状態近傍」... のように、状態近傍を次々に辿ることで、 $S$  の全ての状態に到達できる。  
仮定 3:  $x' \in N(x)$  ならば、 $x \in N(x')$ 。

ただし、近傍に動くことで評価関数が大きく変わってしまうように状態近傍を定義すると、焼きなまし法は機能しない。

巡回セールスマン問題では、「訪問順において任意の隣り合った 2 つの都市を交換したもの」を近傍として定義することが多い。

## 遷移確率 (1)

- はじめに
- ファジ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

遷移確率:  $P(x, x', T)$   
今、状態  $x$  にいるとして、 $x'$  に移るべきかどうかの確率。  
ただし、 $T$  は「温度」。

- ✓  $F(x') < F(x)$  ならば、「状態遷移したほうが得」なので、遷移確率は大きくなる。
- ✓  $F(x') > F(x)$  ならば、「状態遷移したほうが損」なので、遷移確率は小さくなる。特に、 $T \rightarrow 0$  ならば、この場合の遷移確率もゼロに近づく。(逆に言えば、 $T > 0$  のときは、状態遷移したほうが損でも、遷移する確率はゼロでない。)
- ✓  $T$  が高いと、遷移確率は  $1/2$  に近づく。(必須ではない)

## 遷移確率 (2)

- はじめに
- ファジ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

Gibbs 分布:

$$g(x, T) = \frac{1}{\sum_{z \in S} \exp(-F(z)/T)} \cdot \exp(-F(x)/T)$$

- ✓  $T \rightarrow 0$  のとき、 $g(x, T) \rightarrow 0 (x \notin O)$ ,  $g(x, T) \rightarrow 1/N_{opt} (x \in O)$  となる。ただし、 $N_{opt}$  は最適状態の数。
- ✓ 水色部分は  $T$  のみに依存する定数。

遷移確率の例 (ボルツマンの受理関数):

$$P(x, x', T) = B \left( \frac{g(x', T)}{g(x, T)} \right) = \frac{1}{1 + \exp((F(x') - F(x))/T)}$$

ただし、 $B(s) = s/(s + 1)$ 。上記の水色部分は約分されて消える。

## 焼きなまし法のアルゴリズム

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

焼きなまし法のアルゴリズム:

1. まず、初期状態  $x(0)$  を決め、 $k := 0$  とする。
2.  $x(k)$  の近傍  $N(x(k))$  から一点を選び、 $x'(k)$  とする。
3. 繰り返し回数  $k$  によって温度  $T(k)$  を決定する。
4.  $[0, 1]$  の一様乱数を発生し、それが  $P(x(k), x'(k), T(k))$  より小さければ、 $x(k+1) = x'(k)$  に遷移。そうでなければ  $x(k+1) = x(k)$  とし、元の状態に留まる。
5.  $k := k + 1$  とし、あらかじめ決められた繰り返し回数  $k_{max}$  以上ならば、終了。そうでなければ、ステップ 2 に戻る。

- ✓  $T(k)$  は徐々に温度を下げ、最後はゼロになるようにスケジュールされる。
- ✓ 繰り返しの中での一番評価関数が小さい  $x(k)$  が最適解の候補。(最良状態を 1 つだけ記憶しておく)
- ✓ 実際は、温度スケジュール、評価関数の定義、近傍の定義、遷移確率の定義によって結果は大きく変わる。

## マルコフ連鎖 (1)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

SA の収束性を議論するために、マルコフ連鎖の考え方が必要。

- ✓  $x(k)$  を確率変数とみなす。
- ✓  $p(x(t))$  を確率変数  $x(t)$  の確率分布とし、 $S$  の要素の数の長さの行ベクトルである。たとえば、 $n$ -都市の TSP では、 $S$  の要素の数は全ての組み合わせ  $n!$  であるので、 $p(x(t))$  は  $n!$  の長さの行ベクトルになる。 $i$  番目の要素は  $x(t)$  が特定のある状態  $x_i \in S$  にある確率を表し、総和は 1 となる。
- ✓ 未来の挙動  $p(x(t'))$  が現在の  $x(t)$  ( $t' > t$ ) のみによって影響される ( $p(x(t')|x(t)) = p(x(t')|x(\tau), \tau \leq t)$ ) 場合、「マルコフ過程」という。この場合、特に状態数が有限で時間も離散的なので、「マルコフ連鎖」という。
- ✓ 時刻  $k$  において、 $x_i \in S$  から  $x_j \in S$  に遷移する確率を  $m(x_i, x_j, k)$  とする。 $m(x_i, x_j, k)$  を  $(i, j)$  要素とする行列を  $M(k)$  とおく (マルコフ連鎖の遷移確率行列)。各行の要素の総和はおのこの 1 で、 $p(x(k)) = p(x(k-1))M(k-1) = p(x(k-2))M(k-2)M(k-1) = \dots$  となる (チャップマン=コルモゴロフの方程式)。ただし、実際に  $M(k)$  を計算するのは得策ではない。

## マルコフ連鎖 (2)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

$M(k)$  が  $k$  によらないなら、

$$p(x(k+m)) = p(x(k))M^m$$

マルコフ連鎖のエルゴード定理:  
 $k$  によらない遷移確率行列  $M$  が既約かつ非周期的ならば、

$$\lim_{k \rightarrow \infty} pM^k$$

は  $p$  の値によらず、ある決まった値に収束する。

- ✓ 既約 (エルゴード的) とは、任意の  $(i, j)$  に対し、 $m$  が存在し、 $M^m$  の  $(i, j)$  成分が正となること。⇒ 任意の  $x_i$  から任意の  $x_j$  に到達可能。
- ✓ 非周期的とは、 $i$  を決めて、 $M^n$  の  $(i, i)$  成分が正となるような全ての  $n$  の最大公約数が 1。⇒ 周期的の場合は  $x_i$  から  $x_i$  に戻れる時間が最大公約数の倍数しかない。

## 焼きなまし法の収束性の証明

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

仮定:

- ✓ 全ての状態に対し、その近傍の数  $L$  は一定である。⇒  $x_i$  のとき  $x_j$  が選択される確率は、常に  $1/L$
- ✓ 遷移確率として、 $K(g(x', T)/g(x, T))$  を採用。ただし、 $K(\cdot)$  は  $0 \leq K(s) \leq 1$  ( $s \geq 0$ )、 $K(0) = 0$ 、 $K(s) = sK(1/s)$ 。

SA に対し、マルコフ連鎖の遷移確率行列が存在し、既約かつ非周期的となる (証明省略)。よって、ある 1 つの確率分布に収束する。また、

$$P(x, x', T)g(x, T) = P(x', x, T)g(x', T)$$

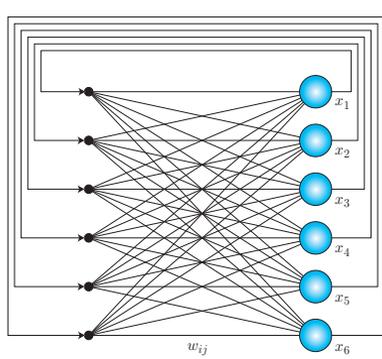
つまり、Gibbs 分布はマルコフ連鎖の平衡状態分布。

焼きなまし法によって、固定した  $T$  に対し、確率分布は Gibbs 分布に収束する。

ゆっくりと  $T \rightarrow 0$  とすることで、大域的最適状態に確率的に収束。

# ボルツマンマシン

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン**
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン**
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略



$w_{i,i} = 0$   
 $w_{i,j} = w_{j,i}$   
 $W = [w_{i,j}]$   
 $W$  は対称行列で  
 対角成分がゼロ  
 ニューロンの出力  $x_i$ :  
 $0$  or  $1$   
 $x_i$  が  $1$  となる **確率**:  

$$\frac{1}{1 + \exp\left(-\frac{\sum w_{i,j}x_j - \theta_i}{T}\right)}$$
 → シグモイド関数が確率を決める

$x_i$  の値の更新は、一度に 1 つのニューロンに対してのみで、そのニューロンはランダムに選ばれる。

# ボルツマンマシンの動き

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン**
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン**
- ボルツマンマシンの動き**
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- ✓ ニューロンの数 (=  $n$ ) の長さのビット列が状態:  $x = (x_1, \dots, x_n)$
- ✓ **擬似エネルギー**

$$E = -\frac{1}{2}xWx^T + x\Theta$$

- を最小化する状態に確率的に収束。
- ただし、 $\Theta = (\theta_1, \dots, \theta_n)^T$
- ✓ **一種の焼きなまし法を実行している。**
- ✓ ビット列の 1 ビットを変えたものが状態近傍
- ✓  $K(\cdot)$  として  $B(s) = s/(s+1)$  を採用した場合に相当。

# 焼きなまし法との等価性 (1)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン**
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン**
- ボルツマンマシンの動き**
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

$x$  の近傍  $x'$  との擬似エネルギーの差:  
 $x$  の  $i$  番目のビットが反転したとする。

- ✓  $0 \rightarrow 1$  の場合:

$$E(x') - E(x) = -\sum_{j=1}^n w_{i,j}x_j + \theta_i$$

- ✓  $1 \rightarrow 0$  の場合:

$$E(x') - E(x) = \sum_{j=1}^n w_{i,j}x_j - \theta_i$$

$w_{i,i} = 0, w_{i,j} = w_{j,i}$  なる式を使った。

# 焼きなまし法との等価性 (2)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン**
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン**
- ボルツマンマシンの動き**
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

遷移確率に、

$$P(x, x', T) = \frac{1}{1 + \exp((E(x') - E(x))/T)}$$

を使った場合、

$$P(x, x', T) = \begin{cases} \frac{1}{1 + \exp\left(-\frac{\sum w_{i,j}x_j - \theta_i}{T}\right)} & (0 \rightarrow 1 \text{ の場合}) \\ \frac{1}{1 + \exp\left(\frac{\sum w_{i,j}x_j - \theta_i}{T}\right)} & (1 \rightarrow 0 \text{ の場合}) \end{cases}$$

## 焼きなまし法との等価性 (3)

一方、ボルツマンマシンにおいて、 $x_i(k+1) = 1$  となる確率は、

$$\frac{1}{1 + \exp\left(-\frac{\sum w_{i,j}x_j - \theta_i}{T}\right)}$$

$x_i(k+1) = 0$  となる確率は、

$$1 - \frac{1}{1 + \exp\left(-\frac{\sum w_{i,j}x_j - \theta_i}{T}\right)} = \frac{1}{1 + \exp\left(\frac{\sum w_{i,j}x_j - \theta_i}{T}\right)}$$

焼きなまし法の遷移確率と一致し、焼きなまし法と等価であることが確認できた。

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

## 自己ループの解消

実は、 $w_{i,i} \neq 0$  の場合でも、条件を満たす等価な重み係数が得られる。

$x_i = 0$  or  $1$  であることに着目すると、 $x_i^2 = x_i$ 。よって、エネルギー関数の  $x_i^2$  の項は  $x_i$  と書ける。

### 疑似エネルギー関数

$$E = -\frac{1}{2}x\bar{W}x^T + x\bar{\Theta}$$

に対し、

$$\bar{W} = W + W_{diag}$$

と分解する。ただし、 $W_{diag}$  は対角成分で、 $W$  はそれ以外。そのとき、

$$E = -\frac{1}{2}xWx^T + x(\bar{\Theta} - (\bar{w}_{1,1}, \dots, \bar{w}_{n,n})/2)$$

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

## 巡回セールスマン問題のエネルギー関数

$N$  都市の場合、 $N \times N$  のニューロン  $x_{i,j}$  ( $i, j = 1, \dots, N$ ) を用意する。 $i$  番目に訪れる都市が  $j$  ならば  $x_{i,j} = 1$  でそれ以外ならばゼロ。

- ✓ 総距離数を  $E_1$  とする。
- ✓  $i$  番目は 1 つの都市のみなので、

$$E_2 = K_2 \sum_{i=1}^N \left(1 - \sum_{j=1}^N x_{i,j}\right)^2$$

- ✓  $j$  番目の都市は 1 回しか訪れないので、

$$E_3 = K_3 \sum_{j=1}^N \left(1 - \sum_{i=1}^N x_{i,j}\right)^2$$

総エネルギー関数は、 $E = E_1 + E_2 + E_3$ 。 $K_2, K_3$  は制約を満たす程度に大きくする。

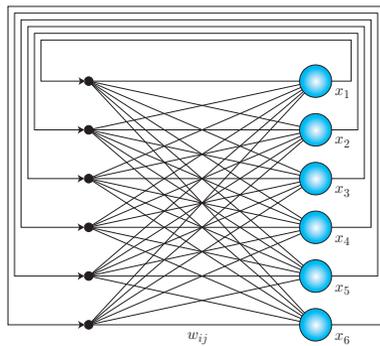
- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- 有限の状態数の場合
- 巡回セールスマン問題
- SA とは
- 状態近傍
- 遷移確率
- SA アルゴリズム
- マルコフ連鎖
- SA の収束性の証明
- ボルツマンマシン
- ボルツマンマシンの動き
- SA との等価性
- 自己ループの解消
- TSP の場合
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

## Hopfield 型ネットワークと連想記憶

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- Hopfield 型ネットワーク
- エネルギー関数
- 連想記憶とは
- パターンを記憶する
- しきい値  $\theta$  の求め方
- Hopfield 型ネットワークによる連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

## Hopfield 型ネットワーク

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶**
- Hopfield 型ネットワーク
- エネルギー関数
- 連想記憶とは
- パターンを記憶する
- しきい値  $\theta$  の求め方
- Hopfield 型ネットワークによる連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略



$w_{i,i} = 0$   
 $w_{i,j} = w_{j,i}$   
 $W = [w_{i,j}]$   
 $W$  は対称行列で  
 対角成分がゼロ

ニューロンの出力  $x_i$ :  
0 or 1

$$x_i = f\left(\sum w_{i,j}x_j - \theta_i\right)$$

$$f(s) = \begin{cases} 1 & (s \geq 0) \\ 0 & (s < 0) \end{cases}$$

$x_i$  の値の更新は、一度に 1 つのニューロンに対してのみで、そのニューロンはランダムに選ばれる。ボルツマンマシンとは違い、ニューロンの選択を除けば**確定的**に動く。

## Hopfield 型ネットワークのエネルギー関数

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶**
- Hopfield 型ネットワーク
- エネルギー関数**
- 連想記憶とは
- パターンを記憶する
- しきい値  $\theta$  の求め方
- Hopfield 型ネットワークによる連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

Hopfield 型ネットワークのエネルギー関数:

$$E = -\frac{1}{2}xWx^T + x\Theta$$

ボルツマンマシンと同じ

ビット列が変化した場合:

$$E(x') - E(x) = -\left|\sum_{j=1}^n w_{i,j}x_j - \theta_i\right|$$

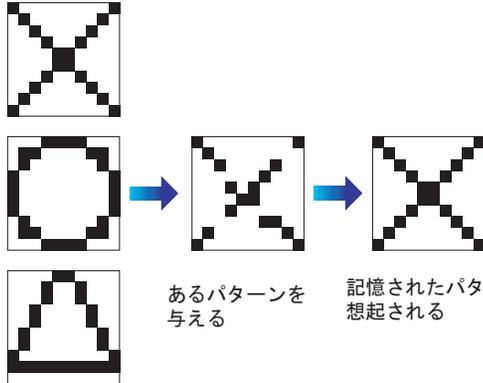
よって  $E(\cdot)$  は単調減少。

- ✓ ボルツマンマシンは、**確率的**に動作し、**焼きなまし法**に等価で、**大域的に収束**。
- ✓ Hopfield 型ネットワークは、**確定的**に動作し、**離散状態での山下り法**と等価で、**極小値に収束**。

## 連想記憶とは

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶**
- Hopfield 型ネットワーク
- エネルギー関数
- 連想記憶とは**
- パターンを記憶する
- しきい値  $\theta$  の求め方
- Hopfield 型ネットワークによる連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

Hopfield 型ネットワークは、連想記憶としても動作する。



あるパターンを与える → 記憶されたパターンが想起される

これらの複数のビット列のパターンを記憶させておく

## パターンを記憶する (1)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶**
- Hopfield 型ネットワーク
- エネルギー関数
- 連想記憶とは
- パターンを記憶する
- しきい値  $\theta$  の求め方
- Hopfield 型ネットワークによる連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

まず、0/1 のビット列を  $\pm 1$  の形に変形する。

$$z = (z_1, \dots, z_N), \quad z_i = 2x_i - 1$$

記憶したいパターン ( $\pm 1$  からなるベクトル):  $\tilde{z}$  とする。このとき、**重み行列**:

$$W = \tilde{z}^T \tilde{z} - I$$

エネルギー関数:

$$E = -\frac{1}{2}zWz^T = -2xWx^T + 2xWe^T - eWe^T$$

$$e = (1, \dots, 1)$$

を考える。 ( $-eWe^T$  は定数になる)

## パターンを記憶する (2)

- はじめに
- フジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- Hopfield 型ネットワーク
- エネルギー関数
- 連想記憶とは
- パターンを記憶する
- しきい値  $\theta$  の求め方
- Hopfield 型ネットワークによる連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

このようにして得られた重み行列は、対称行列かつ対角成分はゼロ。  
( $z_i$  は  $\pm 1$  なので  $z_i^2 = 1$  となるから...)

- ✓  $z = \pm \tilde{z}$  のとき:  
 $zz^T = \pm z\tilde{z}^T = \tilde{z}\tilde{z}^T = N$  なので、

$$E = -\frac{1}{2}zWz^T = -\frac{1}{2}\{(z\tilde{z}^T)(z\tilde{z}^T)^T - zz^T\} = (-N^2 + N)/2$$

- ✓  $z \neq \pm \tilde{z}$  のとき:  
 $zz^T = \tilde{z}\tilde{z}^T = N, |z\tilde{z}^T| = L < N$  なので、

$$E = -\frac{1}{2}zWz^T = -\frac{1}{2}\{(z\tilde{z}^T)(z\tilde{z}^T)^T - zz^T\} = (-L^2 + N)/2$$

$z = \pm \tilde{z}$  のとき、 $E$  は最小値を持つ。

## パターンを記憶する (3)

- はじめに
- フジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- Hopfield 型ネットワーク
- エネルギー関数
- 連想記憶とは
- パターンを記憶する
- しきい値  $\theta$  の求め方
- Hopfield 型ネットワークによる連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

複数のパターン

$$\tilde{z}^{(1)}, \dots, \tilde{z}^{(M)}$$

を記憶する場合を考えよう。

$\tilde{z}^{(k)}$  に対する重み行列・エネルギー関数をそれぞれ  
 $W^{(k)} = (\tilde{z}^{(k)})^T \tilde{z}^{(k)} - I, E^{(k)} = -zW^{(k)}z^T/2$  とすると、求める重み行列は、

$$W = W^{(1)} + \dots + W^{(M)}$$

で、総エネルギー関数は

$$E = E^{(1)} + \dots + E^{(M)}$$

となる。

多くの場合、各々のエネルギー関数の最小値  $\tilde{z}^{(k)}$  が、総エネルギー関数  $E$  の極小値になっていることが期待できる。  
特に、相異なる  $\tilde{z}^{(k)}$  同士が直交している場合、 $E$  は、 $M$  個の最小値  $\tilde{z}^{(k)}$  を持つ。

## しきい値 $\theta$ の求め方

- はじめに
- フジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- Hopfield 型ネットワーク
- エネルギー関数
- 連想記憶とは
- パターンを記憶する
- しきい値  $\theta$  の求め方
- Hopfield 型ネットワークによる連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

$$E = -\frac{1}{2} \sum_{k=1}^M zW^{(k)}z^T$$

$$= 4 \sum_{k=1}^M \left\{ -\frac{1}{2}xW^{(k)}x^T + \frac{1}{2}xW^{(k)}e^T + \frac{1}{4}eW^{(k)}e^T \right\}$$

$$e = (1, \dots, 1)$$

であるから、

$$\Theta = (\theta_1, \dots, \theta_N) = \frac{1}{2} \sum_{k=1}^M eW^{(k)}$$

とすればよい。

## Hopfield 型ネットワークによる連想記憶

- はじめに
- フジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- Hopfield 型ネットワーク
- エネルギー関数
- 連想記憶とは
- パターンを記憶する
- しきい値  $\theta$  の求め方
- Hopfield 型ネットワークによる連想記憶
- 自己組織化ネットワーク
- 連伝的アルゴリズム
- 連伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- ✓ 学習パターンが Hopfield 型ネットワークのエネルギー極小値になっていることを期待
- ✓ Hopfield 型ネットワークは山下り法で、エネルギーは単調減少



初期値に近い極小値 (~ 学習パターンのどれか?) に収束する。

問題点:

- ✓ パターンを記憶する時に、記憶パターン以外にも安定平衡点を生じる場合がある  
→ 偽記憶
- ✓ 記憶できるパターン数:  $0.14N$  より下。

はじめに ファジ理論 ニューラルネットワーク FF型 NN SA とボルツマンマシン Hopfield NN と連想記憶 自己組織化ネットワーク 自己組織化写像 SOM の構造 SOM の学習 SOM の学習結果 格子形状について SOM の使い道 連想的アルゴリズム 連想的プログラミング 多目的最適化と GA 進化戦略	<h2 style="color: #003366;">自己組織化ネットワーク</h2>	2009 年後期 - 69 / 115
ソフトコンピューティング		2009 年後期 - 69 / 115

はじめに ファジ理論 ニューラルネットワーク FF型 NN SA とボルツマンマシン Hopfield NN と連想記憶 自己組織化ネットワーク 自己組織化写像 SOM の構造 SOM の学習 SOM の学習結果 格子形状について SOM の使い道 連想的アルゴリズム 連想的プログラミング 多目的最適化と GA 進化戦略	<h2 style="color: #003366;">自己組織化写像</h2>	2009 年後期 - 70 / 115
ソフトコンピューティング	<ul style="list-style-type: none"> <li>✓ ここでは、自己組織化 (特徴) 写像 (SOM, Self-Organizing Map, Self-Organizing Feature Map, コホーネンの自己組織化ネットワーク) について学ぶ。</li> <li>✓ SOM は、教師なし学習で、 <ul style="list-style-type: none"> <li><math>n</math>-次元入力ベクトル <math>\Rightarrow</math> 2次元メッシュ</li> </ul> </li> <li>の写像を作成する。</li> <li>✓ SOM の用途としては、 <ul style="list-style-type: none"> <li>✗ クラスタリング</li> <li>✗ ベクトル量子化</li> </ul> </li> <li>などがあげられる。</li> </ul>	2009 年後期 - 70 / 115
ソフトコンピューティング		2009 年後期 - 70 / 115

はじめに ファジ理論 ニューラルネットワーク FF型 NN SA とボルツマンマシン Hopfield NN と連想記憶 自己組織化ネットワーク 自己組織化写像 SOM の構造 SOM の学習 SOM の学習結果 格子形状について SOM の使い道 連想的アルゴリズム 連想的プログラミング 多目的最適化と GA 進化戦略	<h2 style="color: #003366;">SOM の構造</h2>	2009 年後期 - 71 / 115
ソフトコンピューティング	<div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> <p>競合層</p> </div> <div> <p>競合層の各ニューロンは、<math>n</math>次元ベクトル</p> <math display="block">y_i = (y_{i,1}, \dots, y_{i,n})</math> <p>を重みとして持つ。→ その「重み」を学習</p> <p>競合層は格子状に並んでおり、ニューロン間の距離が定義される。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><b>勝者ユニット (BMU, Best Matching Unit):</b> 入力ベクトル <math>x = (x_1, \dots, x_n)</math> と一番近い <math>y_{i,j}</math> を持つニューロンが勝者ユニット</p> <math display="block">i^* = \operatorname{argmin}_{i=1, \dots, M} \ x - y_{i,j}\  = \operatorname{argmin}_{i=1, \dots, M} \sqrt{(x_1 - y_{i,1})^2 + \dots + (x_n - y_{i,n})^2}</math> <p><math>M</math>: 競合層のニューロン (= ユニット) の数</p> </div> </div> </div>	2009 年後期 - 71 / 115
ソフトコンピューティング		2009 年後期 - 71 / 115

はじめに ファジ理論 ニューラルネットワーク FF型 NN SA とボルツマンマシン Hopfield NN と連想記憶 自己組織化ネットワーク 自己組織化写像 SOM の構造 SOM の学習 SOM の学習結果 格子形状について SOM の使い道 連想的アルゴリズム 連想的プログラミング 多目的最適化と GA 進化戦略	<h2 style="color: #003366;">SOM の学習 (1)</h2>	2009 年後期 - 72 / 115
ソフトコンピューティング	<ol style="list-style-type: none"> <li>1. まず、競合層の各ニューロンの重みをランダムに初期化。 <math>k := 1</math> とする。</li> <li>2. 入力ベクトル <math>x^{(k)}</math> を入力層に入力し、勝者ユニット <math>y_{i^*}</math> を決定。</li> <li>3. 勝者ユニットおよびその近傍ユニットの重みを <math>x^{(k)}</math> によって修正 <math display="block">y_i^{(k+1)} = y_i^{(k)} + \alpha(k) \gamma^{(k)}(i, i^*) (x^{(k)} - y_i^{(k)})</math> </li> <li>4. 予め決めた繰り返し回数に <math>k</math> が到達していれば終了。そうでなければ、 <math>k := k + 1</math> として、2 に戻る。</li> </ol> <p>ここで、 <math>0 \leq \alpha(k) &lt; 1</math> は学習係数で <math>k</math> に関して単調減少。 <math>\gamma^{(k)}(\cdot, \cdot)</math> は、近傍を定義する写像。</p>	2009 年後期 - 72 / 115
ソフトコンピューティング		2009 年後期 - 72 / 115

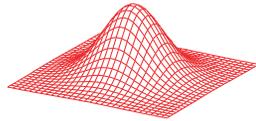
## SOMの学習 (2)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF型 NN
- SAとボルツマンマシン
- Hopfield NNと連想記憶
- 自己組織化ネットワーク
- 自己組織化写像
- SOMの構造
- SOMの学習
- SOMの学習結果
- 格子形状について
- SOMの使い道
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化とGA
- 進化戦略

$\gamma^{(k)}(i, i^*)$  は近傍のみに修正を効かせるための関数。

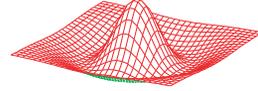
「メッシュ空間での、 $i$  番目のユニットと  $i^*$  番目のユニット (勝者ユニット) の距離」を  $d_{i,i^*}$  とおく。

ガウス基底:



$$\gamma^{(k)}(i, i^*) = \exp(-d_{i,i^*}^2 / (2\sigma_k^2))$$

メキシカンハット関数:



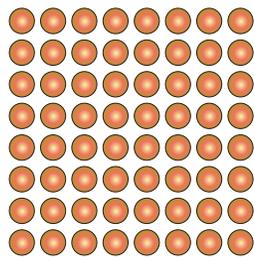
$$\gamma^{(k)}(i, i^*) = (1 - pd_{i,i^*}^2 / \sigma_k^2) \cdot \exp(-d_{i,i^*}^2 / (2\sigma^2))$$

- ✓  $\sigma_k (> 0)$  は  $k$  に関して単調減少。また、 $p > 0$  とする。
- ✓ 近傍ユニットを入力ベクトルに近づけるよう修正する。
- ✓ メキシカンハット関数では、勝者ユニットからある程度離れたユニットに対しては、逆に入力ベクトルから離れるよう修正。

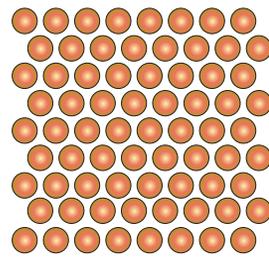
## 格子形状について

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF型 NN
- SAとボルツマンマシン
- Hopfield NNと連想記憶
- 自己組織化ネットワーク
- 自己組織化写像
- SOMの構造
- SOMの学習
- SOMの学習結果
- 格子形状について
- SOMの使い道
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化とGA
- 進化戦略

単なる格子状ではなく、はちのす状の配列が最近良く用いられる。



格子状配列

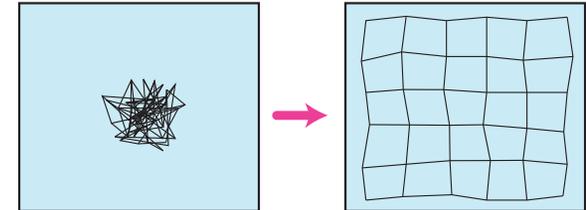


はちのす状配列  
= ハニカム配列

## SOMの学習結果

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF型 NN
- SAとボルツマンマシン
- Hopfield NNと連想記憶
- 自己組織化ネットワーク
- 自己組織化写像
- SOMの構造
- SOMの学習
- SOMの学習結果
- 格子形状について
- SOMの使い道
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化とGA
- 進化戦略

入力ベクトルの範囲に SOM のユニットのベクトルが集まる。



長方形の領域に  
一様に入力ベクトルを加えた場合

## SOMの使い道

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF型 NN
- SAとボルツマンマシン
- Hopfield NNと連想記憶
- 自己組織化ネットワーク
- 自己組織化写像
- SOMの構造
- SOMの学習
- SOMの学習結果
- 格子形状について
- SOMの使い道
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化とGA
- 進化戦略

**自己組織化:** 生物の細胞が集まって組織を作るように、自分自身で自発的に組織や構造をつくり出す機構を自己組織化という。自発的秩序形成。Self-Organization。

SOM では、入力信号のみで勝手に入力信号にフィットするメッシュを作り上げる。

- ✓ 入力ベクトルに勝者ユニットのベクトルを対応させる → (学習) ベクトル量子化 (LVQ)。
- ✓ LVQ は画像圧縮・音声認識などに用いられる。
- ✓ 高次元の入力ベクトルのグループ分けがメッシュの上でなされる。(クラスタリング)

はじめに	<h1>遺伝的アルゴリズム</h1>	ソフトコンピューティング	2009 年後期 - 77 / 115
ファジイ理論			
ニューラルネットワーク			
FF 型 NN			
SA とボルツマンマシン			
Hopfield NN と連想記憶			
自己組織化ネットワーク			
遺伝的アルゴリズム			
遺伝的アルゴリズムとは			
遺伝子と染色体			
遺伝子型と表現型			
GA の流れ			
交叉の方法			
突然変異の方法			
適合度の計算			
淘汰の方法			
GA の問題			
単純 GA			
スキーマ			
スキーマ定理			
積み木仮説			
遺伝的プログラミング			
多目的最適化と GA			
進化戦略			

	<h1>遺伝的アルゴリズムとは</h1>		
はじめに	<div style="border: 1px solid red; padding: 5px;"> <p>遺伝的アルゴリズム (Genetic Algorithm, GA): John Holland によって提案 (1975) された、近似解を探索するメタヒューリスティックアルゴリズム。探索すべきパラメータの組を遺伝子で表現した個体を複数用意し、適応度の高い個体を優先的に選択して交叉・突然変異などの操作を繰り返しながら解を探索する。</p> </div>	ソフトコンピューティング	2009 年後期 - 78 / 115
ファジイ理論			
ニューラルネットワーク			
FF 型 NN			
SA とボルツマンマシン			
Hopfield NN と連想記憶			
自己組織化ネットワーク			
遺伝的アルゴリズム			
遺伝的アルゴリズムとは			
遺伝子と染色体			
遺伝子型と表現型			
GA の流れ			
交叉の方法			
突然変異の方法			
適合度の計算			
淘汰の方法			
GA の問題			
単純 GA			
スキーマ			
スキーマ定理			
積み木仮説			
遺伝的プログラミング			
多目的最適化と GA			
進化戦略			

	<h1>遺伝子と染色体</h1>																										
はじめに	<p>実数値 GA などもあるのだが、ここでは基本的な染色体の構造を示す。</p> <div style="border: 1px solid blue; padding: 5px; margin: 10px 0;"> <p>1 つの染色体 (Chromosome):</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>...</td><td>...</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td> </tr> </table> <p>← 12bit の整数を 0/1 で表現 「2 進数表現」 or 「グレイコード」</p> <p style="margin-left: 150px;">... ..</p> <p style="margin-left: 150px;">↙ 0n/Off の論理値を 1bit で表現</p> </div> <ul style="list-style-type: none"> <li>✓ 0/1 の 1bit の部分を遺伝子 (Gene) と呼ぶ。</li> <li>✓ 複数の遺伝子が集まったものを染色体 (Chromosome) という。</li> <li>✓ ある遺伝子の場所のことを遺伝子座 (Locus) という。</li> <li>✓ 対立遺伝子 (Allele) とは、ある遺伝子座に入るべき遺伝子の種類。上記の場合 0 と 1。</li> <li>✓ 個体 (Individual) とは、染色体によって表現される個。染色体が 0/1 のビット列の遺伝子型のことを言及するのに対し、表現型までも含んだ概念。</li> </ul>	1	0	1	1	0	0	1	0	1	1	0	1	0	0	1	...	...	0	1	0	0	1	1	0	ソフトコンピューティング	2009 年後期 - 79 / 115
1		0	1	1	0	0	1	0	1	1	0	1	0	0	1	...	...	0	1	0	0	1	1	0			
ファジイ理論																											
ニューラルネットワーク																											
FF 型 NN																											
SA とボルツマンマシン																											
Hopfield NN と連想記憶																											
自己組織化ネットワーク																											
遺伝的アルゴリズム																											
遺伝的アルゴリズムとは																											
遺伝子と染色体																											
遺伝子型と表現型																											
GA の流れ																											
交叉の方法																											
突然変異の方法																											
適合度の計算																											
淘汰の方法																											
GA の問題																											
単純 GA																											
スキーマ																											
スキーマ定理																											
積み木仮説																											
遺伝的プログラミング																											
多目的最適化と GA																											
進化戦略																											

	<h1>遺伝子型と表現型</h1>		
はじめに	<ul style="list-style-type: none"> <li>✓ 染色体上の表現 (ビット列) = 遺伝子型</li> <li>✓ 探索空間のパラメータ = 表現型</li> </ul> <div style="border: 1px solid black; padding: 5px; margin: 10px 0; text-align: center;"> <p>表現型を染色体上にコーディングしたものが遺伝子型</p> </div> <p>グレイコード (Gray code): 隣り合う数のハミング距離 (異なるビットの数) が 1</p> <p>バイナリ ⇒ グレイ</p> $g_k = \begin{cases} b_{\ell-1} & (k = \ell - 1) \\ b_{k+1} \vee b_k & (k < \ell - 1) \end{cases}$ <p>グレイ ⇒ バイナリ</p> $b_k = \sum_{i=k}^{\ell-1} g_i \pmod{2}$ <p>ℓ はビット長。グレイコードを用いる理由は後述。</p>	ソフトコンピューティング	2009 年後期 - 80 / 115
ファジイ理論			
ニューラルネットワーク			
FF 型 NN			
SA とボルツマンマシン			
Hopfield NN と連想記憶			
自己組織化ネットワーク			
遺伝的アルゴリズム			
遺伝的アルゴリズムとは			
遺伝子と染色体			
遺伝子型と表現型			
GA の流れ			
交叉の方法			
突然変異の方法			
適合度の計算			
淘汰の方法			
GA の問題			
単純 GA			
スキーマ			
スキーマ定理			
積み木仮説			
遺伝的プログラミング			
多目的最適化と GA			
進化戦略			

## 遺伝的アルゴリズムの流れ

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- GA の問題
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

1. ランダムに染色体を作成し、複数の個体を作成する。個体の集まりのことを、**集団 (Population)** という。
2. 集団からいくつか個体のペア (親: Parent) を選び、**遺伝子同士の交叉 (Crossover)** をして新たな個体 (子: Offspring) を生成する。「子は親と入れ替わる」方法と「親も生き残る」方法がある。
3. 集団の各個体に、ある確率で**突然変異 (Mutation)** をおこす。
4. 集団の各々の個体の**適応度 (Fitness)** を、最適化すべき評価関数から計算する。
5. 適応度が低く、環境に適合しない個体を**淘汰 (Selection)** する。「子は親と入れ替わる」方法の場合は、**集団サイズ (Population size)** (= 集団内の個体数) を最初の値に戻すために、**適応度の高い個体がコピーされる**。「親も生き残る」方法の場合は、**増えた子の数だけ淘汰される**場合が多い。
6. 繰り返し回数が事前に決めた値に達していれば終了。そうでなければ、2 に戻る。

## 交叉の方法

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- GA の問題
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- ✓ **一点交叉法:**  
 親: 1 1 1 0 | 0 1 0      子: 1 1 1 0 | 1 1 1  
 0 0 1 1 | 1 1 1      0 0 1 1 | 0 1 0  
 交叉の場所はランダムに選ばれる
- ✓ **多点交叉法:**  
 親: 1 1 | 1 0 | 0 1 0      子: 1 1 | 1 1 | 0 1 0  
 0 0 | 1 1 | 1 1 1      0 0 | 1 0 | 1 1 1  
 これも、交叉の場所 (上の例では 2 箇所) はランダムに選ばれる。
- ✓ **一様交叉法:**  
 親: 1 1 1 0 0 1 0      子: 1 0 1 0 1 1 0  
 0 0 1 1 1 1 1      0 1 1 1 0 1 1  
 遺伝子ごとにランダム

## 突然変異の方法

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- GA の問題
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- 問題毎に以下のような方法を使い分ける。
- ✓ 一般的な方法: 各遺伝子をランダムに対立遺伝子に置き換え。
  - ✓ 逆位: ランダムに選ばれた 2 点間の要素の順序を逆転。
  - ✓ 転座: ランダムに選ばれた 2 点間の要素を他の位置のものに入れ替え。
  - ✓ 重複: ランダムに選ばれた 2 点間の要素を他の位置にコピー。遺伝子長の変化を許す場合。
  - ✓ 移動: ランダムに遺伝子を 2 個選択して 2 番目の遺伝子を 1 番目の遺伝子の前に移動。
  - ✓ 挿入: 遺伝子を挿入。遺伝子長の変化を許す場合。
  - ✓ 欠失: 遺伝子を消去。遺伝子長の変化を許す場合。
  - ✓ 摂動: 値にランダムな小さな値を加える。実数値 GA などの場合。
- 突然変異率は交叉率より小さく取るのが普通

## 適合度 (適応度) の計算 (1)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- GA の問題
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

評価関数:  $V(x) \rightarrow \max$  ( $x$ : 表現型)

方法によっては、適合度と次世代に生き残る確率は比例する。

適合度の例:

$$\text{Fitness}(x) = \frac{V(x) - V_{\min}}{V_{\max} - V_{\min}} + a$$

$V_{\min}$ : 集団の中の評価関数の最小値  
 $V_{\max}$ : 集団の中の評価関数の最大値

$a$  が大きいと集団の中での適合度の比が小さくなる

## 適合度 (適応度) の計算 (2)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム**
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- GA の問題
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- ✓ 前ページの例で、 $a$  を決める方法として、

$$a = \max \left( 0, \frac{V_{max} + V_{min} - 2V^*}{V_{max} - V_{min}} \right)$$

- がある。ここで、 $V^*$  は  $V(x)$  の平均値。
- ✓ この他、次のようなシグマ切断法もよく使われる。

$$\text{Fitness}(x) = V(x) - (V^* - c\sigma)$$

ここで、 $V^*$  は  $V(x)$  の平均値で、 $\sigma$  は  $V(x)$  の標準偏差。 $c$  は定数。

## 淘汰の方法

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム**
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法**
- GA の問題
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- ✓ ルーレット方式: 適合度に比例する確率で現在の集団から 1 個体づつ選び出す。適合度の高い個体は、2 回以上選ばれる確率が高くなる。確率的に選ぶ方法と、確定的に選ぶ方法 (ドント氏法 = 選挙の比例代表方式) があるが、確率的に選ぶほうが良いとされている。
- ✓ ランク方式: 適合度の値ではなく、その個体が集団中で何番目に適合度が高いかによって選択を行う。あらかじめ、どの順位がいくつ子孫を残せるかを決めておく。
- ✓ トーナメント方式: 組み分けをして、組の中から良い個体を 1 つあるいは 2 つ以上選ぶ。
- ✓ エリート戦略: 集団の中から最も良い適合度を持つ個体をいくつか選び、それらは無条件に次世代に生き残る方式。残りの個体は、ルーレット方式など他の方法によって選ばれる。

## 遺伝子アルゴリズムの問題

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム**
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- GA の問題**
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- ✓ だまし問題の存在: たとえば、“111” が最適で次に “000” が適応度が高い場合を考えよう。“110” や “101” よりも “010” や “100” のほうが評価値が良い場合がある。この場合、“000” に多く個体が集まる。これを「だまし問題」という。“000” を元に最適値 “111” を見つける場合、ハミング距離は 3 であるが、その間にある遺伝子型が評価が悪い状態になっている。(グレイコードを使う理由)
- ✓ 初期収束: 初めのほうの世代で飛びぬけて良い個体が出現すると、その遺伝子が急速に拡散し、集団の多様性が失われてしまう状態。
- ✓ ヒッチハイキング: 比較的良好な個体の中の悪い遺伝子が、良好な個体に「ヒッチハイク」することで、広がってしまう現象。

他に、だまし問題でもないのに GA では求めにくい問題もあり、GA-hard な問題と言われる。

## 単純 GA

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム**
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- 単純 GA**
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

GA の理論的考察が難しいため、以下の単純 GA に限って考察されることが多い。

**単純 GA (Simple GA, SGA):** 以下の組み合わせの GA を単純 GA という

- ✓ 対立遺伝子は 2 つ = 0/1
- ✓ 一点交叉
- ✓ ルーレット選択
- ✓ 通常の突然変異

## スキーマ

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- GA の問題
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

スキーマ (Schema) とは、たとえば以下のようなものである。

$$H = * * 0 * 1 1 * 0 * *$$

ここで、“\*” には “0” か “1” が入る = ワイルドカード。

- ✓ 定義長:  $\delta(H)$ : ワイルドカード以外の文字 (0 か 1) の中で、一番左のものから一番右のものまでの距離。上の例では  $\delta(H) = 5$ 。
- ✓ オーダ (次数):  $O(H)$ : ワイルドカード以外の文字 (0 か 1) の数。

## スキーマ定理 (1)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- GA の問題
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- ✓ 次世代に選択されるスキーマ  $H$  の個体数期待値

$$m(H, k) \frac{f(H, k)}{\bar{f}(k)}$$

- $k$ : 世代数,  $m(H, k)$ : スキーマ  $H$  を持つ個体数,  $f(H, k)$ : スキーマ  $H$  をもつ個体の平均適合度,  $\bar{f}(k)$ : 集団全体の平均適合度
- ✓ スキーマ  $H$  が交叉で破壊されない確率:

$$1 - \frac{p_c \delta(H)}{\ell - 1}$$

- $p_c$ : 交叉率,  $\ell$ : 染色体の長さ
- ✓ スキーマ  $H$  が突然変異で破壊されない確率:

$$(1 - p_m)^{O(H)} \approx 1 - p_m O(H)$$

$p_m$ : 遺伝子あたりの突然変異率

## スキーマ定理 (2)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- GA の問題
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

スキーマ定理 (GA の基本定理):

$$m(H, k + 1) \geq m(H, k) \frac{f(H, k)}{\bar{f}(k)} \cdot \left[ 1 - \frac{p_c \delta(H)}{\ell - 1} - p_m O(H) \right]$$

次世代のスキーマ  $H$  を持つ個体数の見積もり

$p_c > p_m$ ,  $\delta(H) > O(H)$  なので、突然変異の項の影響は小さい。よって、「定義長が小さく」、「 $f(H, k)$  が全体の平均  $\bar{f}(k)$  より大きい」スキーマの個体数が増大することがわかる。

「短くて評価値が良いスキーマが生き残りやすい」

## 積み木仮説

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的アルゴリズムとは
- 遺伝子と染色体
- 遺伝子型と表現型
- GA の流れ
- 交叉の方法
- 突然変異の方法
- 適合度の計算
- 淘汰の方法
- GA の問題
- 単純 GA
- スキーマ
- スキーマ定理
- 積み木仮説
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略

- ✓ 「短くて評価値が良いスキーマ」のことを「積み木」(Building block) という。
- ✓ 「最良個体 (あるいは十分に実用となる準最適な個体) は積み木の組み合わせで得られる」という、希望的観測 = 積み木仮説
- ✓ 積み木仮説が成り立つためには、「表現型に近い個体は遺伝子型も類似している」、「各遺伝子座間の干渉が少ない」という条件が必要がある。

はじめに ファジイ理論 ニューラルネットワーク FF 型 NN SA とボルツマンマシン Hopfield NN と連想記憶 自己組織化ネットワーク 遺伝的アルゴリズム <b>遺伝的プログラミング</b> 遺伝的プログラミング GP における数式表現 GP における S 式表現 GP の交叉 GP の突然変異 EA 等との関連 機械学習 多目的最適化と GA 進化戦略	<h2 style="color: white;">遺伝的プログラミング</h2>	ソフトコンピューティング 2009 年後期 - 93 / 115
--	---	-------------------------------------

はじめに ファジイ理論 ニューラルネットワーク FF 型 NN SA とボルツマンマシン Hopfield NN と連想記憶 自己組織化ネットワーク 遺伝的アルゴリズム <b>遺伝的プログラミング</b> 遺伝的プログラミング GP における数式表現 GP における S 式表現 GP の交叉 GP の突然変異 EA 等との関連 機械学習 多目的最適化と GA 進化戦略	<p><b>遺伝的プログラミング (Genetic Programming, GP):</b> 遺伝的アルゴリズムを拡張したもの。遺伝子型の表現として、<b>配列ではなく木構造を用いる</b>。このため、<b>数式やプログラムのコードなど、構造を持ったデータを表現</b>することができる。</p> <ul style="list-style-type: none"> <li>✓ プログラムを木構造で表現するときは、LISP(プログラミング言語の1つ)のS式を用いることが多い。</li> <li>✓ 関数を木構造で表現するときも、S式-likeな手法をとり、ノードに演算子・関数、定数・変数をとって表現する。</li> <li>✓ 関数を表現するときには、木構造ではなく、逆ポーランド記法を用いて配列にて表現し、GAを適用することもある。</li> </ul> <p>逆ポーランド記法: <math>3 + 2 * x - y</math> は、  <math>3, 2, x, *, +, y, -</math></p> <p>あるいは  <math>3, 2, x, *, y, -, +</math></p> <p>と表現する。</p>	ソフトコンピューティング 2009 年後期 - 94 / 115
--	---	-------------------------------------

はじめに ファジイ理論 ニューラルネットワーク FF 型 NN SA とボルツマンマシン Hopfield NN と連想記憶 自己組織化ネットワーク 遺伝的アルゴリズム <b>遺伝的プログラミング</b> 遺伝的プログラミング GP における数式表現 GP における S 式表現 GP の交叉 GP の突然変異 EA 等との関連 機械学習 多目的最適化と GA 進化戦略	<h2 style="color: white;">GP における数式表現</h2>	ソフトコンピューティング 2009 年後期 - 95 / 115
--	--	-------------------------------------

例:  $3 + 2x - \sin \theta$

- ✓ +, -, \*, sin のノードは非終端記号
- ✓ 2, 3, x, θ のノードは終端記号

はじめに ファジイ理論 ニューラルネットワーク FF 型 NN SA とボルツマンマシン Hopfield NN と連想記憶 自己組織化ネットワーク 遺伝的アルゴリズム <b>遺伝的プログラミング</b> 遺伝的プログラミング GP における数式表現 GP における S 式表現 GP の交叉 GP の突然変異 EA 等との関連 機械学習 多目的最適化と GA 進化戦略	<h2 style="color: white;">GP における S 式表現</h2>	ソフトコンピューティング 2009 年後期 - 96 / 115
--	--	-------------------------------------

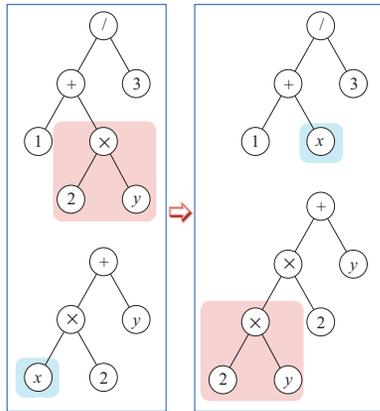
例:

```
(progn (setq x 0) (dotimes (i 10) (setq x (add x i))))
```

## GP の交叉

はじめに  
 ファジ理論  
 ニューラルネットワーク  
 FF 型 NN  
 SA とボルツマンマシン  
 Hopfield NN と連想記憶  
 自己組織化ネットワーク  
 遺伝的アルゴリズム  
 遺伝的プログラミング  
 遺伝的プログラミング  
 GP における数式表現  
 GP における S 式表現  
**GP の交叉**  
 GP の突然変異  
 EA 等との関連  
 機械学習  
 多目的最適化と GA  
 進化戦略

GP の交叉: それぞれの親に対し、1 つのノードをランダムに選び、そのノード以下をそっくり入れ替える。



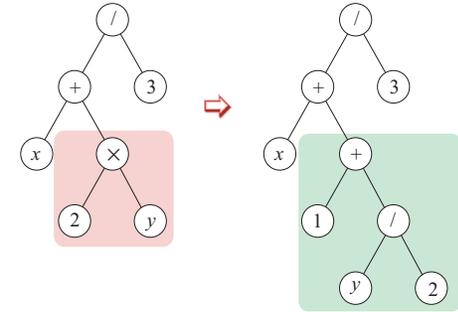
ソフトコンピューティング

2009 年後期 - 97 / 115

## GP の突然変異 (1)

はじめに  
 ファジ理論  
 ニューラルネットワーク  
 FF 型 NN  
 SA とボルツマンマシン  
 Hopfield NN と連想記憶  
 自己組織化ネットワーク  
 遺伝的アルゴリズム  
**遺伝的プログラミング**  
 遺伝的プログラミング  
 GP における数式表現  
 GP における S 式表現  
 GP の交叉  
**GP の突然変異**  
 EA 等との関連  
 機械学習  
 多目的最適化と GA  
 進化戦略

通常の突然変異: あるノード以下をランダムな木構造に入れ替える。



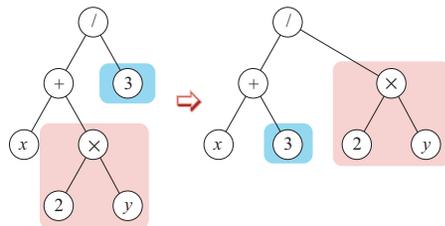
ソフトコンピューティング

2009 年後期 - 98 / 115

## GP の突然変異 (2)

はじめに  
 ファジ理論  
 ニューラルネットワーク  
 FF 型 NN  
 SA とボルツマンマシン  
 Hopfield NN と連想記憶  
 自己組織化ネットワーク  
 遺伝的アルゴリズム  
 遺伝的プログラミング  
 遺伝的プログラミング  
 GP における数式表現  
 GP における S 式表現  
 GP の交叉  
**GP の突然変異**  
 EA 等との関連  
 機械学習  
 多目的最適化と GA  
 進化戦略

逆位: 2 つのノードをランダムに選び、そのノード以下をそっくり入れ替える。



ソフトコンピューティング

2009 年後期 - 99 / 115

## 進化的アルゴリズムなどとの関連

はじめに  
 ファジ理論  
 ニューラルネットワーク  
 FF 型 NN  
 SA とボルツマンマシン  
 Hopfield NN と連想記憶  
 自己組織化ネットワーク  
 遺伝的アルゴリズム  
**遺伝的プログラミング**  
 遺伝的プログラミング  
 GP における数式表現  
 GP における S 式表現  
 GP の交叉  
 GP の突然変異  
**EA 等との関連**  
 機械学習  
 多目的最適化と GA  
 進化戦略

- ✓ 進化的計算 (Evolutionary Computation): 進化の機構にヒントを得ている、計算手法の総称。
- ✓ 進化的アルゴリズム (Evolutionary Algorithm, EA): 進化の機構にヒントを得ている、個体群ベースのメタヒューリスティックな最適化アルゴリズムの総称。進化的計算の一分野。

よって、

[進化的計算] ⊃ [進化的アルゴリズム]  
 ⊃ [GA, GP, 進化戦略 (ES), 進化的プログラミング]

ソフトコンピューティング

2009 年後期 - 100 / 115

## 機械学習としての GP,GA

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 遺伝的プログラミング
- GP における数式表現
- GP における S 式表現
- GP の交叉
- GP の突然変異
- EA 等との関連
- 機械学習
- 多目的最適化と GA
- 進化戦略

- ✓ たとえば IF-THEN ルールのようなもので表現された、機械の行動ルールを学習させる方法、いわゆる「機械学習」について考える。
- ✓ 機械学習の場合、適合度を計算する場合、決まった評価関数が与えられているケースは少ない。1 世代の適合度として、特定の初期値・環境におけるタスク成功の報酬を考えることが多い。つまり、毎世代、適合度関数が変化する。
- ✓ ミシガンアプローチでは、**1 つのルールが 1 個体**として考えられる。GA/GP というより強化学習の枠組みで考えられる場合が多い。同じ個体ばかり増えないように、同じ個体のコピーを作る代わりに「信頼度」というパラメータを考える。
- ✓ ピッツアプローチ (ピッツバーグアプローチ) では、**ルールの集合体が 1 個体**である。GA/GP の枠組みでは自然な方法である。

## 多目的最適化と遺伝的アルゴリズム

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 多目的最適化問題とは
- 解の候補間の優越関係
- パレート最適解
- 多目的最適化手法
- GA による多目的最適化
- 最終的な解の選択
- 進化戦略

## 多目的最適化と遺伝的アルゴリズム

## 解の候補間の優越関係

この節では、ベクトル同士の比較として、つぎのような記法を用いる。

$p$  次元ベクトル  $a = (a_1, \dots, a_p)$ ,  $b = (b_1, \dots, b_p)$  に対し、

$$a \leq b \iff a_i \leq b_i \quad (i = 1, \dots, p)$$

$$a \leq b \iff a_i \leq b_i \quad (i = 1, \dots, p), \text{ かつ } a_k < b_k \quad \exists k$$

$$a < b \iff a_i < b_i \quad (i = 1, \dots, p)$$

$x^1, x^2 \in \mathcal{G}$  に対し、

✓  $F(x^1) \leq F(x^2)$  ならば「 $x^1$  は  $x^2$  に優越する」という。

✓  $F(x^1) < F(x^2)$  ならば「 $x^1$  は  $x^2$  に強い意味で優越する」という。

「 $x^1$  は  $x^2$  に優越する」 $\Rightarrow x^2$  より  $x^1$  が良い解の候補。

## 多目的最適化問題とは

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 多目的最適化問題とは
- 解の候補間の優越関係
- パレート最適解
- 多目的最適化手法
- GA による多目的最適化
- 最終的な解の選択
- 進化戦略

多目的最適化とは、制約条件

$$g_1(x) \leq 0, \dots, g_m(x) \leq 0$$

の下で、**複数の評価関数**

$$F_1(x), \dots, F_p(x)$$

をできるだけ最小化するような  $x \in \mathcal{R}^n$  を求める問題。

$F_i$  同士で、トレードオフの関係になっている場合があり、**完全最適解が存在する保証はない。**

ここで、解の制約領域を

$$\mathcal{G} = \{x | g_1(x) \leq 0, \dots, g_m(x) \leq 0\}$$

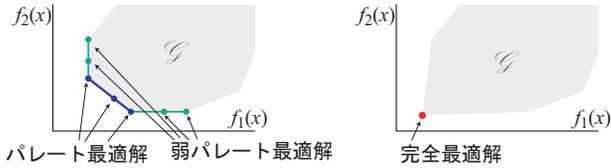
とおく。また、評価関数ベクトルを  $F = (F_1(x), \dots, F_p(x))$  とする。

## パレート最適解

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 多目的最適化問題とは
- 解の候補間の優越関係
- パレート最適解
- 多目的最適化手法
- GA による多目的最適化
- 最終的な解の選択
- 進化戦略

**定義:**

- ✓  $x^0 (\in \mathcal{G})$  に強い意味で優越する  $x \in \mathcal{G}$  が存在しないとき、 $x^0$  を弱パレート最適解という。
- ✓  $x^0 (\in \mathcal{G})$  に優越する  $x \in \mathcal{G}$  が存在しないとき、 $x^0$  を(強)パレート最適解という。
- ✓  $x^0 (\in \mathcal{G})$  が他の全ての  $x \in \mathcal{G}$  に優越するとき、 $x^0$  を完全最適解という。



少なくともパレート最適解の1つ、できればパレート最適解集合そのものを求めたい。

## 多目的最適化手法

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 多目的最適化問題とは
- 解の候補間の優越関係
- パレート最適解
- 多目的最適化手法
- GA による多目的最適化
- 最終的な解の選択
- 進化戦略

✓ 重み係数による方法: 重み係数  $w_i (> 0)$  を考えて、

$$F_a = w_1 F_1(x) + \dots + w_p F_p(x) \rightarrow \min$$

パレート最適解の1つが求まる。また、像  $F(\mathcal{G})$  が凸集合ならば、 $w$  を色々変えることにより、全てのパレート最適解が得られる。実際には、重み係数  $w$  の選び方が難しい。

✓  $\epsilon$  制約法: 1つだけ評価関数を選び(説明の簡単のためそれを  $F_1(x)$  とする)、残りを制約とみなす。

$$F_1(x) \rightarrow \min, \quad \text{subject to } x \in \mathcal{G}, F_2(x) \leq \epsilon_2, \dots, F_p(x) \leq \epsilon_p$$

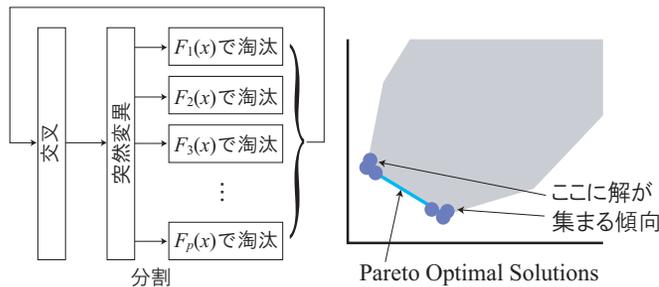
✓  $\epsilon_k$  を色々変えることにより、全てのパレート最適解が得られる。  
 ✓ 遺伝的アルゴリズムによる多目的最適化: 本節で説明。

## GA による多目的最適化 (1)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 多目的最適化問題とは
- 解の候補間の優越関係
- パレート最適解
- 多目的最適化手法
- GA による多目的最適化
- 最終的な解の選択
- 進化戦略

**Schaffer** による方法: 通常の GA とほとんど同じであるが、

- ✓ 適応度計算・淘汰においては、集団を  $p$  分割し ( $p$ : 評価関数の個数)、各部分集団ごとに異なる評価関数を適用する。
- ✓ 交叉・突然変異に関しては、集団を1つにまとめたものに対して行う。



## GA による多目的最適化 (2)

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 多目的最適化問題とは
- 解の候補間の優越関係
- パレート最適解
- 多目的最適化手法
- GA による多目的最適化
- 最終的な解の選択
- 進化戦略

**Goldberg** の方法: 適応度計算は全ての評価関数に対して行っておく。その上で**ランキング選択**を用いる。ランクは以下のように決定する。

1. 他に優越されない個体群をランク1とする。
2. ランク1の個体を除いた集団中で、他に優越されない個体群をランク2とする。
3. ランク1と2の個体を除いた集団中で、他に優越されない個体群をランク3とする。
4. 以下、繰り返し。

**Fonseca** らの方法: Goldberg の方法と同様に、ランキング選択を用いる。ただし、ランクの付け方が異なる。  
**ランクの付け方** 個体  $X_i$  が  $n_i$  個の個体に優越されているとき、

$$\text{rank}(X_i) = 1 + n_i$$

最終的に得られた集団の中で、他に優越されない個体群が「パレート最適解集合」を近似。

## 最終的な解の選択

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 多目的最適化問題とは
- 解の候補間の優越関係
- パレート最適解
- 多目的最適化手法
- GA による多目的最適化
- 最終的な解の選択
- 進化戦略

得られたパレート最適解から、1つ「妥協解」を選び出したい。  
重心法: パラメータ空間 ( $x$  の空間) で、パレート最適解集合 (の近似) の重心を計算し、それに一番近い解を「妥協解」とする。

## 進化戦略

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略
- 進化戦略
- ( $\mu, \lambda$ )-ES の流れ
- 突然変異
- 組み換え

## 進化戦略

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略
- 進化戦略
- ( $\mu, \lambda$ )-ES の流れ
- 突然変異
- 組み換え

進化戦略 (進化的戦略, Evolution Strategy, ES) とは、

- ✓ 実数値ベクトル  $x \in \mathbb{R}^n$  の最適化

$$F(x) \rightarrow \min$$

- ✓ を行うメタヒューリスティック手法。
- ✓ GA とくらべて、交叉 (ES では組み替えという) よりも突然変異が主な探索手段。
- ✓ 色々な方法があるが、ここで述べるのは ( $\mu, \lambda$ )-ES。  $\mu$  は親の数で、  $\lambda$  は子の数。「親は生き残らない」方法に相当。
- ✓ 「親も生き残る」方法は ( $\mu + \lambda$ )-ES と呼ばれる。

## ( $\mu, \lambda$ )-ES の流れ

- はじめに
- ファジイ理論
- ニューラルネットワーク
- FF 型 NN
- SA とボルツマンマシン
- Hopfield NN と連想記憶
- 自己組織化ネットワーク
- 遺伝的アルゴリズム
- 遺伝的プログラミング
- 多目的最適化と GA
- 進化戦略
- 進化戦略
- ( $\mu, \lambda$ )-ES の流れ
- 突然変異
- 組み換え

( $\mu, \lambda$ )-ES のおおまかな流れは以下の通り。

1. 各個体のランダムな初期化。
2. 組み換えという操作により、  $\mu$  個の親より  $\lambda$  個の子を生成。ただし、  $\lambda \geq \mu$  で、親は死滅。
3.  $\lambda$  個の個体それぞれに突然変異させる。
4.  $\lambda$  個の個体それぞれに評価値  $F(x)$  を計算。
5.  $\lambda$  個の個体より、最良の  $\mu$  個を選択。
6. あらかじめ決められた繰り返し回数に達していなければ、2. に戻る。さもなければ終了。

一般に、  $\lambda$  は  $\mu$  の 7 倍程度が良いとされている。

## 突然変異 (1)

はじめに  
ファジイ理論  
ニューラルネットワーク  
FF 型 NN  
SA とボルツマンマシン  
Hopfield NN と連想記憶  
自己組織化ネットワーク  
遺伝的プログラミング  
多目的最適化と GA  
進化戦略  
進化戦略  
( $\mu, \lambda$ )-ES の流れ  
突然変異  
組み換え

各個体は、最適化すべきベクトル  $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$  の他に、突然変異の標準偏差  $\sigma = (\sigma_1, \dots, \sigma_n)^T \in \mathbb{R}^n$ 、角度パラメータ

$\alpha = (\dots, \alpha_{ij}, \dots)^T \in \mathbb{R}^{n(n-1)/2}$  ( $i = 1, \dots, n-1, j = i+1, \dots, n$ ) を持つ。

すなわち、各個体は  $(x, \sigma, \alpha)$  で表現される。

突然変異は次の 2 段階からなる。

✓  $\sigma, \alpha$  の突然変異:

$$\sigma_{i,new} = \sigma_{i,old} \exp(\tau' \xi + \tau \xi_i)$$

$$\alpha_{ij,new} = \alpha_{ij,old} + \beta \xi_{ij}$$

$\xi, \xi_i, \xi_{ij}$  は全て平均 0 分散 1 の正規乱数。また、 $\tau, \tau', \beta$  は定数で、Schwefel による推薦値は、

$$\tau = \left(\sqrt{2\sqrt{n}}\right)^{-1}, \quad \tau' = \left(\sqrt{2n}\right)^{-1}, \quad \beta = 0.0873[\text{rad}] = 5[\text{deg}]$$

## 突然変異 (2)

はじめに  
ファジイ理論  
ニューラルネットワーク  
FF 型 NN  
SA とボルツマンマシン  
Hopfield NN と連想記憶  
自己組織化ネットワーク  
遺伝的プログラミング  
多目的最適化と GA  
進化戦略  
進化戦略  
( $\mu, \lambda$ )-ES の流れ  
突然変異  
組み換え

✓  $x$  の突然変異:

- ✗ 各  $i$  に対し、平均値 0、分散  $\sigma_i^2$  の正規乱数  $\eta_i$  を生成し、ベクトル  $\eta = (\eta_1, \dots, \eta_n)^T \in \mathbb{R}^n$  を作る。
- ✗ 行列  $R(\alpha_{ij})$  を、

$$r_{ii} = r_{jj} = \cos \alpha_{ij}, \quad r_{ij} = -r_{ij} = -\sin \alpha_{ij}, \quad r_{kk} = 1 \quad (k \neq i, j)$$

のように作る。他の行列要素は 0 である。

- ✗ ベクトル  $\zeta = (\zeta_1, \dots, \zeta_n)^T \in \mathbb{R}^n$  を

$$\zeta = \prod_{i=1}^n \prod_{j=i+1}^n R(\alpha_{ij}) \eta$$

のように作る。

- ✗  $x$  を、 $x_{new} = x_{old} + \zeta$  のように更新。

✓ 以上を各個体に適用。

## 組み換え

はじめに  
ファジイ理論  
ニューラルネットワーク  
FF 型 NN  
SA とボルツマンマシン  
Hopfield NN と連想記憶  
自己組織化ネットワーク  
遺伝的プログラミング  
多目的最適化と GA  
進化戦略  
進化戦略  
( $\mu, \lambda$ )-ES の流れ  
突然変異  
組み換え

組み換え (recombination) は、GA の交叉に相当する操作である。

たとえば、以下のような操作を行う。

- ✓ 実数全体を対立遺伝子とみなした GA における一様交叉
- ✓ 内分点を任意に取る。
- ✓ 中間点を取る。
- ✓ 成分ごとに内分点を任意に取る。

これらは  $x$  だけでなく、 $(x, \sigma, \alpha)$  全体に関して実行する。